NPS ARCHIVE
1967
LEIPOLD, F.

SOLUTION OF TWO-DIMENSIONAL HEAT PROBLEMS USING THE ALTERNATING DIRECTION METHOD

FREDERICK JAMES LEIPOLD

[redacted]

SOLUTION OF TWO-DIMENSIONAL HEAT PROBLEMS USING THE ALTERNATING DIRECTION METHOD

by

Frederick James Leipold
Lieutenant, United States Navy
B.S.E.E., Purdue University, 1961

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
WITH MAJOR IN MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
September 1967

NPS ARCHIVE
1967
EIPOLD, F.

ABSTRACT

Finite difference approximations to the one-dimensional heat equation $U_t = U_{xx}$ are used to introduce explicit and implicit difference equations. The convergence and stability of these equations is discussed and these concepts are used to show the restrictions imposed on explicit equations. The Implicit Alternating Direction (IAD) method is introduced as a means for solution of the two-dimensional heat equation. Although the IAD method in its basic form is applicable only to parabolic problems, it is possible by slight modification to apply the method to elliptic problems. Two examples are used to illustrate the use of the IAD method for solution of parabolic and elliptic equations for a rectangular region. These examples include a work requirement comparison with other difference methods. A third example is given to show the applicability of the IAD method to non-rectangular regions, in this case a parabolic problem over a circular region. Results of these examples show that the IAD method is a convenient and accurate method when applied to both parabolic and elliptic partial differential equations and suggest applicability to a wide range of problems.

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

TABLE OF CONTENTS

SECTION	TITLE	PAGE
	Abstract	2
	List of Tables	5
	List of Illustrations	7
	Acknowledgements	9
	Introduction	11
I.	Comparison of Explicit and Implicit Difference Equations for the Problem $U_t = U_{xx}$	14
II.	The Explicit and Implicit Difference Equations for the Problem in two Space Variables	26
III.	The IAD Method	28
	Example 1. Solution of a Time-Dependent Problem Using the IAD Method	32
	Example 2. Solution of a Steady State Problem Using the IAD Method	41
	Example 3. Solution of a Time-Dependent Problem in a Non-rectangular Region Using the IAD Method	51
	Conclusions	60
	Bibliography	62

LIST OF TABLES

TABLE	TITLE	PAGE
1	Numerical Solution and Percent Difference at Point (0.6.0.6) for Different Values of Δt and Δx. (Example 1)	38
2	Calculated Values of the Iteration Parameter ρ_p (Example 2)	43
3	Grouped Values of ρ_p (Example 2)	44
4	Analytic and Numerical Solutions for Time 0.1 (Example 3)	57
5	Results of Solutions Using Various Δr, $\Delta \theta$, and Δt at the Point (0.5.45°) (Example 3)	58

LIST OF ILLUSTRATIONS

Figure	Title	Page
1	Flowchart for Program IADMT	65
2	Flowchart for Algorithm Subroutine	67
3	Solution Printout for Program IADMT, Example 1	68
4	Solution Printout for Program IADMT, Example 1	69
5	Plot of Percent Error Vs. $\Delta t + (\Delta x)^2$ for Time 0.005 at Point (.6,.6) IAD Solution of Problem $U_{xx} + U_{yy} = U_t$	70
6	Flowchart for Program IADM	71
7	Solution Printout for Program IADM, Example 2	73
8	Solution Printout for Program IADM, Example 2	74
9	Solution Printout for Program IADM, Example 2	75
10	Solution Printout for Program IADM, Example 2	76
11	Solution Printout for Program IADM, Example 2	77
12	Solution Printout for Program IADM, Example 2	78
13	Solution Printout for Program IADM, Example 2	79
14	Solution Printout for Program IADM, Example 2	80
15	Solution Printout for Program IADM, Example 2	81
16	Solution Printout for Program IADM, Example 2	82
17	Solution Printout for Program IADM, Example 2	83
18	Solution Printout for Program IADM, Example 2	84
19	Solution Printout for Program IADM, Example 2	85
20	Solution Printout for Program IADM, Example 2	86
21	Flowchart for Program IADMC	87
22	Solution Printout for Program IADMC, Example 3	89
23	Plot of Difference Vs. $\Delta t + (\Delta r)^2 + (\Delta\theta)^2$ IAD Solution of Problem $U_{rr} + \frac{1}{r}U_r + \frac{1}{r^2}U_{\theta\theta} = U_t$	90

LIST OF ILLUSTRATIONS

Figure	Title	Page
24	Temperatures in a Square Plate	32
25	Grid for the Region Shown in Figure 24	33
26	Temperatures at the Face of an Infinite Prism	41
27	Temperatures at the Face of an Infinite Cylinder	51
28	Grid for the Region Shown in Figure 27	51

ACKNOWLEDGMENT

I wish to thank Professor U. R. Kodres of the Naval Postgraduate School for his advice and assistance in the preparation of this Thesis.

INTRODUCTION

$U_{xx} + U_{yy} = \frac{1}{k}U_t$ is the parabolic partial differential equation governing the flow of heat in two space dimensions. The constant k appearing in this equation is the diffusivity of the material through which heat flows. We assume here and subsequently that the time unit is chosen such that k = 1. Numerical approximations to the solution of this equation may be obtained by solving an associated difference equation. Depending upon the time level at which the derivatives U_{xx} and U_{yy} are approximated, the difference equation will be one of two types: (1) explicit, which is simple to solve since it involves only one unknown quantity in terms of several known quantities, but which may also require an enormous number of calculations due to limitations on the size of the time-step; and (2) implicit, which does not restrict the size of the time-step, but which involves five unknown quantities and is most easily solved by an iteration process.

A modified implicit difference method, called the Implicit Alternating Direction (IAD) method, was introduced by Peaceman and Rachford in 1955 and may be used to obtain numerical approximations to the solution of the heat flow equation in a direct and convenient manner. In addition, the IAD method may be applied, with very slight modification in form, to elliptic partial differential equations. The IAD method involves two difference equations: the first in which three unknown values of U in the x direction are expressed in terms of three known values of U in the y direction; and the second in which three unknown values of U in the y direction are expressed in terms of three known values of U in the x direction. Each of these equations gives rise to a system of simultaneous equations. However,

the coefficient matrix of each system is tri-diagonal, and therefore the solution is relatively easy to obtain.

In this paper we will first consider the simple one-dimensional heat equation $U_{xx} = U_t$. We will derive the associated explicit and implicit difference equations and then introduce the concepts of convergence and stability to show that a restriction on the time-step for the explicit equation is necessary. The explicit and implicit difference equations for the two-dimensional heat flow equation will be given and the associated restrictions and solution difficulties will be considered.

We shall then examine the difference equations which define the IAD method, showing how these equations are solved and how they are applied to the elliptic problem governed by Laplace's equation $U_{xx} + U_{yy} = 0$. We will discuss some aspects of the proof that the IAD method is both convergent and stable, particularly in regard to selection of an iteration parameter for use in solving elliptic equations.

Three examples are given. The first example is an application of the IAD method to the parabolic partial differential equation in two space variables, and the second is an application to the related steady state problem. The region for the problems in both of these examples is rectangular. The third example is an application of the IAD method to the same parabolic equation, written in polar coordinates, over a circular region. In Examples one and two a comparison is made of the work requirement for the IAD method and the work requirement for other difference methods. Data obtained in Examples

one and three are used in an attempt to obtain bounds for the discretization error.

I. Comparison of Explicit and Implicit Difference Equations for the Problem $U_t = U_{xx}$.

For simplicity, consider the parabolic partial differential equation (PDE)

(1) $U_t(x,t) = U_{xx}(x,t)$ for $a < x < b$, $t > 0$

with boundary and initial conditions

$$U(a,t) = g(t), \quad U(b,t) = h(t), \quad U(x,0) = f(x).$$

This is the equation that specifies the conduction of heat in one space dimension.

To attempt the solution of this problem by a finite difference method, we establish a network of grid points throughout the region $a \le x \le b$, $t > 0$. In the x direction divide the region into strips of width $\Delta x = (b-a)/M$, and in the t direction into strips of width $\Delta t = T/N$, where M and N are arbitrary positive integers, and T is some fixed time for which the solution is desired. We may now use indices i and k to denote the grid points $i\Delta x$ and $k\Delta t$.

If we assume that U(x,t) possesses a sufficient number of partial derivatives, then the relation between U(x,t) and $U(x + \Delta x, t + \Delta t)$ is given by the Taylor's series expansion

$$U(x+\Delta x, t+\Delta t) = U(x,t) + \left[\Delta x \frac{\partial}{\partial x} + \Delta t \frac{\partial}{\partial t}\right] U(x,t) + \frac{1}{2!}\left[\Delta x \frac{\partial}{\partial x} + \Delta t \frac{\partial}{\partial t}\right]^2 U(x,t) + \cdots + \frac{1}{(n-1)!}\left[\Delta x \frac{\partial}{\partial x} + \Delta t \frac{\partial}{\partial t}\right]^{n-1} U(x,t) + R_n,$$

where the remainder term is

$$R_n = (1/n!)\left[\Delta x \frac{\partial}{\partial x} + \Delta t \frac{\partial}{\partial t}\right]^n U(x + \gamma \Delta x, t + \alpha \Delta t)$$

for $0 \le \gamma \le 1$, $0 \le \alpha \le 1$. This leads to the following definition:

Definition. If, in the relation between U(x,t) and $U(x + \Delta x, t + \Delta t)$, we neglect the remainder R_n, then we introduce an error

(called the truncation error) which is of order $(\Delta x + \Delta t)^n$, denoted $0(\Delta x + \Delta t)^n$. By this we mean that there exists a positive constant C such that

$$|R_n| \leq C(\Delta x + \Delta t)^n \text{ as both } \Delta x \text{ and } \Delta t \text{ go to zero.}$$

Now if we let $v_{i,k}$ be the finite difference approximation to $U(x,t)$ at the grid point $(i\Delta x, k\Delta t)$, then by using Taylor's expansion we obtain the following difference approximations for the derivatives U_t and U_{xx}

$$v_t = (v_{i,k+1} - v_{i,k})/\Delta t + 0(\Delta t)$$

$$v_{xx} = (v_{i+1,k} - 2v_{i,k} + v_{i-1,k})/(\Delta x)^2 + 0(\Delta x)^2.$$

Using these approximations, one finite difference approximation to the PDE is

$$(v_{i,k+1} - v_{i,k})/\Delta t = (v_{i-1,k} - 2v_{i,k} + v_{i+1,k})/(\Delta x)^2.$$

If we define $\varphi = (\Delta x)^2/\Delta t$ and then solve for $v_{i,k+1}$, we get the equation

$$(2) \quad v_{i,k+1} = (1/\varphi)v_{i-1,k} + (1-2/\varphi)\, v_{i,k} + (1/\varphi)\, v_{i+1,k}.$$

The boundary and initial conditions are approximated by

$v_{0,k+1} = g\left[(k+1)\Delta t\right]$, $v_{M,k+1} = h\left[(k+1)\Delta t\right]$, and

$v_{i,0} = f(i\Delta x)$, where $i = 0$ and $i = M$ represent the points $x = a$ and $x = b$ respectively, and $k = 0$ represents $t = 0$. Equation (2) is an explicit difference equation since we can calculate the values of $v_{i,k+1}$ explicitly at time level $(k + 1)\Delta t$ if we know the values of $v_{i,k}$ at time $k\Delta t$. Thus by starting with the boundary and initial conditions, equation (2) can be used to calculate the values of v at all grid points for any time $t > 0$.

If we approximate the derivative U_{xx} at time level $(K+1)\Delta t$ rather than at time level $k\Delta t$, we obtain the difference approximation

$$(v_{i,k+1} - v_{i,k})/\Delta t = (v_{i-1,k+1} - 2v_{i,k+1} + v_{i+1,k+1})/(\Delta x)^2.$$

Again, letting $\rho = (\Delta x)^2/\Delta t$ and rearranging, we have

$$(3) \quad v_{i-1,k+1} - (2+\rho)v_{i,k+1} + v_{i+1,k+1} = -\rho v_{i,k}$$

Equation (3) is an implicit difference equation since it involves three unknown quantities at time level $(k+1)\Delta t$ in terms of one known quantity at time $k\Delta t$. The boundary and initial conditions have the same approximations as for the explicit equation. Thus we have a more complicated scheme in this case since at any time level $(k+1)\Delta t$, the implicit equation (3) is written once for each point $1 \leq i \leq M-1$, resulting in the following system of M-1 simultaneous equations in the M-1 unknowns $v_{i,k+1}$,

$$\begin{aligned}
(1+2/\rho)v_{1,K+1} - 1/\rho\, v_{2,K+1} &= v_{1,K} + 1/\rho\, g[(K+1)\Delta t] \\
-1/\rho\, v_{1,K+1} + (1+2/\rho)v_{2,K+1} - 1/\rho\, v_{3,K+1} &= v_{2,K} \\
-1/\rho\, v_{i-1,K+1} + (1+2/\rho)v_{i,K+1} - 1/\rho\, v_{i+1,K+1} &= v_{i,K} \\
-1/\rho\, v_{M-2,K+1} + (1+2/\rho)v_{M-1,K+1} &= v_{M-1,K} + 1/\rho\, h[(K+1)\Delta t]
\end{aligned}$$

Now let $a_i = c_i = -1/\rho$ and $b_i = (1+2/\rho)$, $(i = 1,2,\ldots,M-1)$, and call the known terms on the right hand sides of these equations $d_1, d_2, \ldots, d_{M-1}$. Then after dropping the subscripts k+1, the system has the following form

$$\begin{aligned}
b_1v_1 + c_1v_2 &= d_1 \\
a_2v_1 + b_2v_2 + c_2v_3 &= d_2 \\
a_iv_{i-1} + b_iv_i + c_iv_{i+1} &= d_i \\
a_{M-2}v_{M-3} + b_{M-2}v_{M-2} + c_{M-2}v_{M-1} &= d_{M-2} \\
a_{M-1}v_{M-2} + b_{M-1}v_{M-1} &= d_{M-1}
\end{aligned}$$

The matrix of coefficients for this system is tri-diagonal, and the use of Gaussian elimination results in the following algorithm for solution.

Let $\beta_1 = b_1$, $\gamma_1 = d_1/\beta_1$

$$\beta_i = b_i - (a_i c_{i-1})/\beta_{i-1} \ , \quad \gamma_i = (d_i - a_i \gamma_{i-1})/\beta_i \ , \quad (i = 2, 3, \cdots, M-1) \ ;$$

then $V_{M-1} = \gamma_{M-1}$, $V_i = \gamma_i - (c_i V_{i+1})/\beta_i$, $(i = M-2, M-3, \cdots, 1)$.

Thus we have our choice of two finite difference schemes with which we can attempt to find an approximate solution to the PDE (1). We shall see that one of the conditions required to make the approximation a useful one is a restriction on the size of the time-step Δt when using the explicit equation (2). It will be shown in Example 1 that the use of the implicit equation (3) requires less work than the use of (2), even with the added complexity of solving systems of equations since there is no restriction on the size of Δt in the implicit method.

Whether or not the solution $v_{i,k}$ of the difference equation is a good approximation to the solution $U(x,t)$ of the PDE can be answered by considering the questions of convergence and stability. We begin with the following definitions:

Definition. The difference $w_{i,k} = U(i\Delta x, k\Delta t) - v_{i,k}$ is called the discretization error at the point $(i\Delta x, k\Delta t)$.

Definition. We say that the solution $v_{i,k}$ of the difference equation converges to the solution $U(x,t)$ of the PDE at the point $(i\Delta x, k\Delta t)$ if $\lim_{\Delta t, \Delta x \to 0} w_{i,k} = 0$. If $v_{i,k}$ converges to $U(x,t)$ at every grid point $(i\Delta x, k\Delta t)$, then we say that the difference procedure is convergent. When we speak of convergence, we are speaking about the difference between the exact solution to the PDE and the exact solution to the

difference equation. Since we will generally attempt to solve the difference equation using a digital computer, we cannot expect to obtain the exact solution. Instead we obtain a numerical solution which differs from the exact solution due to round-off error. When this difference remains small, we say that the corresponding procedure is stable.

Before giving a formal definition of stability, we examine the question of round-off error in more detail. Suppose $v(x,t)$ is the exact solution to the difference equation and suppose we replace $v(x_o, t_o)$ by $v(x_o,t_o) + e$ at the grid point (x_o, t_o). This may be considered to be the effect of round-off error or possibly a mistake in the calculations. We call e the error at the point (x_o, t_o). If we then continue the calculation using the value $v(x_o, t_o) + e$, without introducing new errors, we obtain a solution $v^*(x,t)$ at the point (x,t). We call the difference $v^*(x,t) - v(x,t)$ the departure of the solution caused by the error e at (x_o, t_o). If errors are committed at more than one point, we speak of the cumulative departure due to these errors. For linear problems we can use the principle of superposition to conclude that the departure due to two or more errors is the sum of the departures due to the individual errors. We can now give the following definition:

Definition. A difference procedure is stable if the cumulative departure, due to errors in the solution, tends to zero as the maximum of these errors tends to zero and does not "grow faster" than some positive power of $1/\Delta x$ as Δx tends to zero. We say that the cumulative departure does not "grow faster" than some positive power of $1/\Delta x$ to exclude exponential growth in $1/\Delta x$.

The general method used to show convergence is to find a bound for the discretization error $w(x,t)$, and then show that this bound depends on Δx and Δt and tends to zero as $\Delta x, \Delta t \to 0$. Stability is most often proved by a method due to Von Neumann. This method involves expanding the error in a Fourier series and showing by means of the coefficients of this series that this error does not grow as the solution progresses. For the two difference schemes which we have introduced, we will obtain the conditions for convergence and stability by a method given in Forsythe and Wasow [3].

We assume, for convenience, that the associated boundary conditions for the PDE (1) are $g(t) \equiv 0$ and $h(t) \equiv 0$. Without loss of generality we can assume that $a = 0$ and $b = \pi$ since we can modify this interval by means of a linear transformation. Then the exact solution to (1) is

$$(4) \quad U(x,t) = \sum_{r=1}^{\infty} a_r e^{-r^2 t} \sin rx,$$

where the initial function $f(x)$ has been expanded in the convergent Fourier sine series

$$f(x) = \sum_{r=1}^{\infty} a_r \sin rx, \text{ where } a_r = \frac{2}{\pi} \int_0^{\pi} f(x) \sin rx \, dx.$$

Now consider the following difference equation

$$(5) \quad (v_{i,k+1} - v_{k,k})/\Delta t = \sigma(v_{i+1,k+1} - 2v_{i,k+1} + v_{i-1,k+1})/(\Delta x)^2 + (1-\sigma)(v_{i+1,k} - 2v_{i,k} + v_{i-1,k})/(\Delta x)^2,$$

where σ is a parameter. Note that for $\sigma = 0$, equation (5) reduces to the explicit difference equation (2); and for $\sigma = 1$, it reduces to the implicit equation (3). We can now examine the stability and convergence of equations (2) and (3).

Theorem. When $\sigma < 1/2$, equation (5) represents a convergent and

stable difference scheme if $\varphi = (\Delta x)^2/\Delta t \geq 2(1-2\sigma)$. When $\sigma \geq 1/2$, equation (5) represents a convergent and stable difference scheme for all values of φ.

Proof. Suppose that $v(x,t)$ is a solution to (5) which satisfies the conditions: $v(x,0) = f(x)$, $0 < x < \pi$; $v(0,t) = v(\pi,t) = 0$, $0 < t \leq T$. Here, for convenience of notation, we use (x,t) to mean the grid point $(i\Delta x, k\Delta t)$. Assume that $f(x)$ can be written in the Fourier series

$$f(x) = \sum_{r=1}^{\infty} a_r \sin rx, \text{ where}$$

$$a_r = \frac{2}{\pi}\int_0^{\pi} f(x) \sin rx \, dx \text{ and } \sum_{r=1}^{\infty} |a_r| < \infty$$

We now seek a solution of the form

$$v(x,t) = \sum_{r=1}^{\infty} a_r \psi_r(t) \sin rx.$$

If $\psi_r(0) = 1$, the solution satisfies the initial condition; the boundary conditions are automatically satisfied. If we substitute $\psi_r(t) \sin rx$ into (5), we get

$$\psi_r(t+\Delta t)\left(1 + \frac{4\sigma}{\varphi}\sin^2\frac{r\Delta x}{2}\right) = \psi_r(t)\left[1 - \frac{4}{\varphi}(1-\sigma)\sin^2\frac{r\Delta x}{2}\right]$$

With the initial condition $\psi_r(0) = 1$, this difference equation has the solution $\psi_r(t) = \varphi_r^{t/\Delta t}$

where

$$\varphi_r = \frac{1 - \frac{4}{\varphi}(1-\sigma)\sin^2\frac{r\Delta x}{2}}{1 + 4\frac{\sigma}{\varphi}\sin^2\frac{r\Delta x}{2}}$$

and where $t/\Delta t$ takes only positive integer values. So we have as a solution

$$(6) \qquad v(x,t) = \sum_{r=1}^{\infty} a_r \varphi_r^{t/\Delta t} \sin rx,$$

if this series converges. Since $\sum_{r=1}^{\infty} |a_r| < \infty$, a sufficient

condition for convergence (in fact, for uniform convergence) is $|\varphi_r| \leq 1$ for all r. But this means

$$-1 \leq \frac{1 - 4/\rho(1-\sigma)\sin^2 r\frac{\Delta x}{2}}{1 + \frac{4\sigma}{\rho}\sin^2 r\frac{\Delta x}{2}} \leq 1 .$$

The right hand inequality is always satisfied. The left hand inequality is satisfied when $2(1 - 2\sigma) \leq \rho$. Note that this is always the case when $\sigma \geq 1/2$, so that $|\varphi_r| \leq 1$ for any ρ.

To show convergence it is necessary to show that the series (6) tends to that given by (4) as $\Delta t, \Delta x \to 0$.

In the expression for φ_r we replace $\sin^2 r\frac{\Delta x}{2}$ by $r^2(\Delta x)^2/4 + 0(\Delta x)^4$ to get

$$\varphi_r = \frac{1 - r^2\Delta t + \Delta t\left[\sigma r^2 + 4(1-\sigma)O(\Delta x)^2\right]}{1 + \Delta t\left[\sigma r^2 + 4\sigma O(\Delta x)^2\right]} .$$

We now examine the $\lim_{\Delta x, \Delta t \to 0} \varphi_r^{t/\Delta t}$. If we consider this as an iterated limit, we have two cases. These are:

Case 1. $\lim_{\Delta t \to 0} \lim_{\Delta x \to 0} \varphi_r^{t/\Delta t} = \lim_{\Delta t \to 0}\left[\frac{1 - r^2\Delta t + \sigma r^2\Delta t}{1 + \sigma r^2 \Delta t}\right]^{t/\Delta t}$

$$= \lim_{\Delta t \to 0}\left[1 - r^2\Delta t + \frac{\sigma r^4(\Delta t)^2}{1 + \sigma r^2\Delta t}\right]^{t/\Delta t}$$

$$= \left\{\lim_{\Delta t \to 0}\left[1 + \Delta t + \frac{\sigma r^2(\Delta t)^2}{1 + \sigma r^2\Delta t}\right]^{1/\Delta t}\right\}^{-r^2 t} = e^{-r^2 t}$$

Case 2. $\lim_{\Delta x \to 0} \lim_{\Delta t \to 0} \varphi_r^{t/\Delta t}$. As it stands, $\lim_{\Delta t \to 0} \varphi_r^{t/\Delta t}$ is an indeterminate form so we write $\lim_{\Delta t \to 0} \varphi_r^{t/\Delta t} = \lim_{\Delta t \to 0} e^{\frac{t}{\Delta t} \ln \varphi_r}$

$$= e^{t \lim_{\Delta t \to 0} \frac{\ln \varphi_r}{\Delta t}} .$$

Using L'Hospital's rule we have

$$t \lim_{\Delta t \to 0} \frac{\ln \varphi_r}{\Delta t} = t \lim_{\Delta t \to 0} \frac{d}{d\Delta t} \ln \varphi_r = t\left[4(1-2\sigma)O(\Delta x)^2 - r^2\right].$$

So $\lim_{\Delta x\to 0}\lim_{\Delta t\to 0}\varphi_r^{t/\Delta t} = \lim_{\Delta x\to 0} e^{t[4(1-2\sigma)O(\Delta x)^2 - r^2]} = e^{-r^2 t}$

Thus we see that regardless of the way in which Δx and Δt tend to zero, $\lim_{\Delta t,\Delta x\to 0}\varphi_r^{t/\Delta t} = e^{-r^2 t}$ so that every term of (6) tends to the corresponding term of (4).

Since the convergence of (6) is uniform, it is possible to consider the limit, as $\Delta x, \Delta t$ tend to zero, in a termwise fashion. Therefore if $2(1 - 2\sigma) \leq \varphi$, then $\lim_{\Delta t,\Delta x\to 0} v(x,t) = U(x,t)$, and the difference equation (5) represents a convergent scheme.

To show stability it is more convenient to represent the values of $v(x,t)$ at each grid point by means of a finite Fourier series rather than the infinite series (6). This is based on the theory of trigonometric interpolation and uses the orthogonality relations

$$\sum_{r=0}^{M-1} \begin{Bmatrix}\cos\\ \sin\end{Bmatrix} nr\Delta x \begin{Bmatrix}\cos\\ \sin\end{Bmatrix} mr\Delta x = 0 \quad \text{for } n \neq m,\ n,m < M$$

and

$$\sum_{r=0}^{M-1} \cos^2 nr\Delta x = \sum_{r=0}^{M-1} \sin^2 nr\Delta x = M/2,\ 0 < n < M,$$

where $M = \pi/\Delta x$ is an integer. Using these relations, we can write

$$f(x) = \sum_{n=1}^{M-1} A_n \sin nx,$$

where

$$A_n = (2/M) \sum_{r=1}^{M-1}{}' f(r\Delta x) \sin nr\Delta x.$$

This relation will be valid for the points $x = r\Delta x, (r = 1,2,\ldots,M-1)$. We use a finite series since round-off error cannot be regarded as a smooth function independent of Δx.

Now, let $M = \pi/\Delta x$ be the number of grid points on a line for constant t and consider an arbitrary line of errors $e_p (p=1,2,\ldots,M-1)$

at $t = 0$. We then consider these errors as some initial function and represent them by

$$e_s = \sum_{r=1}^{M-1} A_r \sin sr\Delta x \qquad (s = 1,2,\ldots,M-1)$$

where

$$A_r = (2/M) \sum_{p=1}^{M-1} e_p \sin pr\Delta x. \qquad (r = 1,2,\ldots,M-1)$$

Then by the same argument that resulted in the infinite series (6), we show that

$$(7)\quad v^*(s\Delta x, n\Delta t) = \sum_{r=1}^{M-1} A_r \varphi_r^{t/\Delta t} \sin sr\Delta x,$$

is the solution of (5) at the points of the grid for the initial values e_s, or v^* represents the departure due to e_s for the line $t = 0$.

We now determine a bound for v^*. Substituting the expression for A_r in (7) we have

$$(8)\quad v^*(s\Delta x, n\Delta t) = \sum_{p=1}^{M-1} g_{p,s,n} e_p$$

where

$$g_{p,s,n} = (2/M) \sum_{r=1}^{M-1} \varphi_r^{t/\Delta t} \sin sr\Delta x \sin pr\Delta x$$

(Here, our interchange of summations is valid since these are finite series). If $2(1 - 2\sigma) \leq \varphi$, then $|\varphi_r| \leq 1$ and $|g_{p,s,n}| \leq 2$. Let $|e_r| \leq \delta$ for $r = 1,2,\ldots,M-1$ where δ represents the maximum of the errors. Then from (8) we have

$$v^*(s\Delta x, n\Delta t) \leq 2M\delta = 2\pi\delta/\Delta x$$

which means that the cumulative departure due to errors is of order $\delta(\Delta x)^{-1}$ which by our definition means that the process is stable.

Therefore, we have proved that for $\varphi \geq 2(1 - 2\sigma)$, equation (5) represents a convergent and stable finite difference scheme. It

follows then that for $\sigma \geq 1/2$, (5) represents a convergent and stable scheme for any positive value of ρ. For $\sigma = 0$, (5) reduces to equation (2) and the condition for convergence and stability is $\rho \geq 2$, or $\Delta t \leq (\Delta x)^2/2$. As previously stated, the explicit method is restrictive in the size of the time-step. For $\sigma = 1$, (5) reduces to the implicit equation (3) and is unrestricted as to the size of the time-step allowed.

The method used to prove stability in the previous theorem is essentially the method of Von Neumann. This method can be extended to problems involving two or more space variables.

Note that for the difference equations examined, the conditions for stability and convergence are identical. Lax and Richtmyer [5] have developed a theory for a large class of linear partial differential equations with constant coefficients which, for slightly different definitions of convergence and stability, states that stability of a difference equation implies convergence of the equation. Briefly, this theory assumes that the initial functions of the linear differential problem belong to some Banach space B, of functions of x. Then associated with every initial function in B is a solution to the difference problem (this solution being a function of x alone when t is fixed) as well as a so-called genuine solution to the PDE. Both of these functions belong to B. Under these conditions a finite difference procedure is called convergent if the solution to the associated difference equation converges to a genuine solution for all initial functions in B. This is a more restrictive definition than the one we have used since we have established convergence for initial functions in some incomplete

subset of the space B. A difference procedure is called stable in this sense if the solution $v(x,t)$, corresponding to some initial function $f(x)$, satisfies

$$||v(x,t)|| \leq M(t) \; || \; f(x) || \quad , \; 0 \leq t \leq T, \text{ for all } f(x) \text{ in } B,$$

where $M(t)$ is independent of Δx. This definition is also more restrictive than ours since we allow a bound which depends on Δx. There is also a difference in the concept of distance between two functions. We have used the norm

$$||v(x,t) - U(x,t)|| = \sup_{a \leq x \leq b} |v(x,t) - U(x,t)| \quad ,$$

whereas Lax uses the mean square norm

$$\left\{ \int_a^b [v(x,t) - U(x,t)]^2 dx \right\}^{1/2}$$

If the definitions of convergence and stability are taken in the above sense, then convergence and stability are proven to be equivalent concepts in the Equivalence Theorem of Lax. It can be shown that explicit difference equation (2), with the restrictions on Δt, actually does satisfy the Lax-Richtmyer theory.

II. The Explicit and Implicit Difference Equations for the Problem in Two Space Variables.

For the problem $U_{xx} + U_{yy} = U_t$, we use the same difference approximations for U_{xx} and U_t and a similar approximation for U_{yy} (the approximations for U_{xx} and U_{yy} taken at time level $k\,\Delta t$). We change notation slightly, letting the subscripts i and j refer to increments in the x and y directions, and the superscript k refer to increments in time. The explicit difference equation is

$$(9) \qquad V_{i,j}^{k+1} = V_{i,j}^{k} + \frac{1}{\varphi}\left[V_{i-1,j}^{k} + V_{i+1,j}^{k} + V_{i,j-1}^{k} + V_{i,j+1}^{k} - 4V_{i,j}^{k}\right]$$

Again taking the approximation for U_{xx} and U_{yy} at time level $(k+1)\,\Delta t$ results in the implicit difference equation

$$(10) \qquad V_{i-1,j}^{k+1} + V_{i+1,j}^{k+1} + V_{i,j-1}^{k+1} + V_{i,j+1}^{k+1} - (4+\varphi)V_{i,j}^{k+1} = -\varphi V_{i,j}^{k}$$

As in the case of the difference equations in one space variable, we find that the explicit procedure (9) is both stable and convergent for certain values of φ and the implicit procedure (10) is stable and convergent for all values of φ. (Here we have taken $\varphi = (\Delta x)^2/\Delta t = (\Delta y)^2/\Delta t$.) The restriction on φ for the explicit equation is $\varphi \geq 4$. This result is proven in Douglas [6], in which the difference equations are treated as matrix operators so that the result is valid for any number of space dimensions.

For the explicit equation, $\varphi \geq 4$, means that $\Delta t \leq (\Delta x)^2/4$, so that the allowable time-step is half the size of that allowed for the explicit equation in one space variable. This may mean that an enormous number of calculations are required to obtain a solution for a particular time T.

Equation (10) again results in a system of simultaneous equations which can be solved by Gaussian elimination. However, such a method for solution is impractical as there are now five unknowns per equation and the number of required computations is more than twice the number required for the tri-diagonal system. The solution of this equation is possible by iteration techniques, but we shall demonstrate the solution of the PDE in two space variables by means of a modified implicit method called the Implicit Alternating Direction (IAD) method.

III. The IAD Method

We now introduce the IAD method of Peaceman and Rachford [7]. If only one of the second partial derivatives for the PDE, $U_{xx} + U_{yy} = U_t$, say U_{xx}, is replaced by a difference approximation at time $(k + 1)\Delta t$, and U_{yy} is replaced by an approximation at time level $k\Delta t$, then we again have a difference equation with only three unknowns, and this equation may be solved by the use of the previously stated algorithm. The difference equation obtained is said to be implicit in the x direction. We then repeat the process for a second time-step of equal size, reversing the level at which the second derivatives are approximated, to obtain a difference equation implicit in the y direction. When writing these equation, we use time levels $2k + 1$ and $2k + 2$ to emphasize that the time-steps must be of equal size. The equations are:

Implicit in the x direction

$$(11)\quad \frac{V_{i,j}^{2k+1} - V_{i,j}^{2k}}{\Delta t/2} = \frac{V_{i-1,j}^{2k+1} - 2V_{i,j}^{2k+1} + V_{i+1,j}^{2k+1}}{(\Delta x)^2} + \frac{V_{i,j-1}^{2k} - 2V_{i,j}^{2k} + V_{i,j+1}^{2k}}{(\Delta y)^2}$$

Implicit in the y direction

$$(12)\quad \frac{V_{i,j}^{2k+2} - V_{i,j}^{2k+1}}{\Delta t/2} = \frac{V_{i-1,j}^{2k+1} - 2V_{i,j}^{2k+1} + V_{i+1,j}^{2k+1}}{(\Delta x)^2} + \frac{V_{i,j-1}^{2k+2} - 2V_{i,j}^{2k+2} + V_{i,j+1}^{2k+2}}{(\Delta y)^2}$$

We choose $\Delta x = \Delta y$, and let $\varphi = (\Delta x)^2/\Delta t = (\Delta y)^2/\Delta t$. Then, rearranging, we obtain a form more convenient for calculation:

$$(11a)\quad -V_{i-1,j}^{2k+1} + 2(\varphi+1)V_{i,j}^{2k+1} - V_{i+1,j}^{2k+1} = V_{i,j-1}^{2k} + 2(\varphi-1)V_{i,j}^{2k} + V_{i,j+1}^{2k}$$

$$(12a)\quad -V_{i,j-1}^{2k+2} + 2(\varphi+1)V_{i,j}^{2k+2} - V_{i,j+1}^{2k+2} = V_{i-1,j}^{2k+1} + 2(\varphi-1)V_{i,j}^{2k+1} + V_{i+1,j}^{2k+1}$$

Each of these equations leads to $M - 1$ equation in $M - 1$ unknowns at each half time-step, $\Delta t/2$. The systems each have a tri-diagonal coefficient matrix.

In addition to the time-dependent problem $U_{xx} + U_{yy} = U_t$, the IAD method may also be used in the solution of steady state problems; for example, Laplace's equation, $U_{xx} + U_{yy} = 0$. The procedure is essentially one of iteration with each stage of iteration being regarded as a time-step in a time-dependent problem. A starting value is used which approximates the boundary conditions and this serves as an initial condition. Equations (11a) and (12a) are used as alternate stages of iteration with ρ serving as an iteration parameter. We will show by means of examples how the IAD method is used in the solution of both time-dependent and steady state problems.

We now state the following theorem without proof.

Theorem. The IAD method, as applied to the partial differential equations $\Delta U(x,y,t) = U_t(x,y,t)$ and $\Delta U(x,y) = 0$, having either Dirichlet or Von Neumann boundary conditions, is both convergent and stable for any region.

The proof of this theorem can be found in Forsythe and Wasow [3], pp 272 and 411, and in Birkhoff and Varga [8]. We now state some of the results of the proof given in [3] for the IAD method as applied to the steady state problem.

Consider the difference approximations for U_{xx} and U_{yy} used in equations (11) and (12). If we regard the solution of the difference equation as a column vector $V(x,y)$, then the symmetric difference expression used to approximate U_{xx} can be regarded as a matrix A^h operating on V. Thus,

$$A^hV(x,y) = -V(x-\Delta x,y) + 2V(x,y) - V(x+\Delta x,y).$$

Similarly, for U_{yy}, the analogous matrix is A^v, and

$$A^v V(x,y) = -V(x,y-\Delta y) + 2V(x,y) - V(x,y+\Delta y).$$

For the case of Laplace's equation in a square region with n^2 interior grid points, A^h will be an $n^2 \times n^2$ matrix which may be partitioned into an n x n block diagonal matrix with each diagonal block being an n x n tri-diagonal matrix. A^v will be similar to such a matrix. If we use the subscript k to denote the iteration number, the IAD method is then defined by the equations

$$(13) \quad V_{k-1/2} = V_{k-1} - \frac{1}{\varphi_k}\left(A^h V_{k-1/2} + A^v V_{k-1}\right)$$

$$(14) \quad V_k = V_{k-1/2} - \frac{1}{\varphi_k}\left(A^h V_{k-1/2} + A^v V_k\right)$$

where the parameter $\varphi_k = \frac{(\Delta x)^2}{\Delta t_k}$ may vary from iteration to iteration. In effect, we are solving the elliptic (steady state) problem as the asymptotic solution of the time-dependent problem; thus the Δt appearing in the expression for φ_k. In the actual solution process for the elliptic problem, however, Δt will not enter into the method for determining optimal values of φ.

We now define the error E_k as the difference $V_k - U$, where U is the solution of the PDE at each grid point and may thus be regarded as a vector. Then, after eliminating $V_{K-1/2}$ between (13) and (14), the effect of one complete iteration is

$$E_k = P(\varphi_K) E_{k-1},$$

where the matrix operator P is

$$P(\varphi) = \left(I + \tfrac{1}{\varphi} A^v\right)^{-1}\left(I - \tfrac{1}{\varphi} A^h\right)\left(I + \tfrac{1}{\varphi} A^h\right)^{-1}\left(I - \tfrac{1}{\varphi} A^v\right)$$

and I is the identity matrix.

Now, as shown by Forsythe and Wasow, a sufficient condition for convergence is that the spectral radius of $P(\varphi)$ be less than one, where the spectral radius is the maximum of the absolute values

of the eigenvalues of P. This is shown to be true for any positive constant value of φ.

If the region of interest for the PDE is a rectangle, specific statements can be made about the values of the iteration parameter φ_k which lead to the most rapid convergence. As shown in Birkhoff and Varga [8], however, this is not possible in non-rectangular regions. Suppose, for convenience, that the boundary of the region is a square of side length π. Then A^h and A^v commute and, by a theorem due to Frobenius, both matrices share the eigenvectors

$$\sin px \sin qy, \quad (p,q = 1,2,\ldots,M-1),$$

with eigenvalues

$$4 \sin^2 p\,\Delta x/2, \; 4 \sin^2 q\,\Delta y/2, \quad (p,q = 1,2,\ldots,M-1).$$

The eigenvalues of $P(\varphi)$ are then

$$(15) \qquad \frac{(\varphi - 4\sin^2 p\frac{\Delta x}{2})(\varphi - 4\sin^2 q\frac{\Delta y}{2})}{(\varphi + 4\sin^2 p\frac{\Delta x}{2})(\varphi + 4\sin^2 q\frac{\Delta y}{2})}, \quad (p,q = 1,2,\cdots,M-1)$$

From this expression it can be seen that the eigenvalues of $P(\varphi)$ are less than one in absolute value for any positive value of φ. Peaceman and Rachford [7] use the Von Neumann method of stability analysis to derive the same expression as an amplification factor for error. In Example 2 we will show how this expression is used to determine the optimal values of the iteration parameter φ.

We will show in Examples 1 and 2 how the IAD method can be used to obtain an approximate solution to the time-dependent and the steady state problems for a rectangular region. In Example 3 the method will be applied to time-dependent problem over a section of a circular region.

Example 1. Solution of a Time-Dependent Problem Using the IAD Method.

Problem: To find the temperature distribution for various times in a square plate with edge-length one.

The temperature distribution in a square plate of edge-length one (See Figure 24) is governed by the following parabolic partial differential equation:

$$U_{xx}(x,y,t) + U_{yy}(x,y,t) = U_t(x,y,t) ,$$

with boundary conditions

$$U(1,y,t) = U(x,1,t) = 0 \qquad U_x(0,y,t) = U_y(x,0,t) = 0$$

and initial condition

$$U(x,y,0) = 1$$

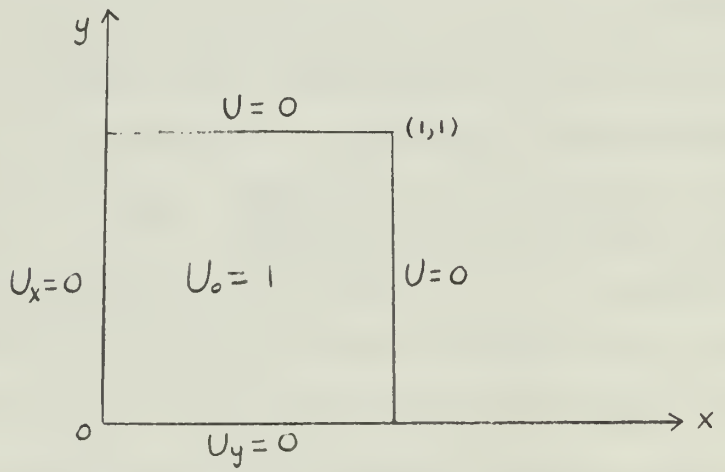

Figure 24 Temperatures in a Square Plate

To solve this problem using the IAD method we use the difference equations

$$(11a) \quad -V_{i-1,j}^{2K+1} + 2(\varphi+1)V_{i,j}^{2K+1} - V_{i+1,j}^{2K+1} = V_{i,j-1}^{2K} + 2(\varphi-1)V_{i,j}^{2K} + V_{i,j+1}^{2K}$$

$$(12a) \quad -V_{i,j-1}^{2K+2} + 2(\varphi+1)V_{i,j}^{2K+2} - V_{i,j+1}^{2K+2} = V_{i-1,j}^{2K+1} + 2(\varphi-1)V_{i,j}^{2K+1} + V_{i+1,j}^{2K+1}$$

where $$\varphi = \frac{(\Delta x)^2}{\Delta t} = \frac{(\Delta y)^2}{\Delta t}$$

Equation (11a) is said to be implicit in the x direction and equation (12a) is said to be implicit in the y direction.

To apply these equations, we divide the region shown in Figure 24 into N increments of length 1/N in both the x and y directions. Here N is an arbitrary positive integer. The resulting grid is shown in Figure 25. Thus $v_{i,j}^{2k+1}$ refers to the difference approximation to the solution U(x,y,t) at the grid point $(i\Delta x, j\Delta y)$ at time $(2k+1)\Delta t$, where $\Delta x = \Delta y = 1/N$.

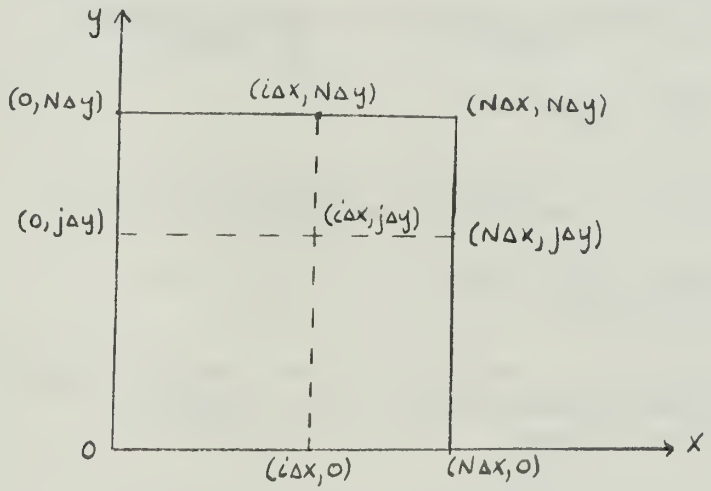

Figure 25 Grid for the Region Shown in Figure 24

Equations (11a) and (12a) can be used for all interior points of the grid and, of course, the values of U are known on the boundaries x = 1 and y = 1. However, at the boundaries x = 0 and y = 0, the boundary conditions are $U_x(0,y,t) = 0$ and $U_y(x,0,t) = 0$, so we must find an approximation involving U_x for U_{xx} at x = 0. and an approximation involving U_y for U_{yy} at y = 0. Since the procedure is the same at both boundaries, the details are given for the boundary x = 0 only.

We assume, for a fictitious point, a distance Δx outside the boundary, that $U(x-\Delta x,y,t) = U(x+\Delta x,y,t)$. This is intuitively

acceptable since $U_x=0$ and the region under consideration can be thought of as one quarter of a square plate having edge-length two and being centered at the origin. For this larger plate (with boundaries held at temperature zero) there would be complete symmetry and hence no heat flow across the x and y axes. We now write the Taylor expansions for $U(x-\Delta x,y,t)$ and $U(x+\Delta x,y,t)$ in terms of $U(x,y,t)$, where (x,y,t) represents the boundary point $[0,j\Delta y,(2k+1)\Delta t]$. Thus we have

$$U(x-\Delta x,y,t) = U(x,y,t) - \Delta x U_x(x,y,t) + \frac{(\Delta x)^2}{2!}U_{xx}(x,y,t) - \frac{(\Delta x)^3}{3!}U_{xxx}(x,y,t) + \ldots.$$

and

$$U(x+\Delta x,y,t) = U(x,y,t) + \Delta x U_x(x,y,t) + \frac{(\Delta x)^2}{2!}U_{xx}(x,y,t) + \frac{(\Delta x)^3}{3!}U_{xxx}(x,y,t) + \ldots.$$

Now, adding these equations, and neglecting terms involving U_{xxxx} and all higher derivatives, we get

$$U(x-\Delta x,y,t) + U(x+\Delta x,y,t) = 2U(x,y,t) + (\Delta x)^2 U_{xx}(x,y,t) + O(\Delta x)^4$$

But $U(x-\Delta x,y,t) = U(x+\Delta x,y,t)$, so we can write

$$(16)\quad U_{xx}(x,y,t) = \frac{2}{(\Delta x)^2}\left[U(x+\Delta x,y,t) - U(x,y,t)\right] + O(\Delta x)^2.$$

$$(17)\quad U_{yy}(x,y,t) = \frac{2}{(\Delta y)^2}\left[U(x,y+\Delta y,t) - U(x,y,t)\right] + O(\Delta y)^2.$$

For notation in the computer program IADMT, we use TST(I,J) to denote the values of our approximation to U(x,y,t) at the end of a first half time-step (using equation 11a) and TEND(I,J) to denote the approximation at the end of a second half time-step (using equation 12a). Thus TEND(I,J) represents the numerical solution at the end of a complete time-step. In the equations that follow, we abbreviate TST(I,J) by T*(I,J) and TEND(I,J) by T(I,J).

All computer programs were run on the IBM 360 computer using the FORTRAN IV language. Since FORTRAN IV does not allow zero subscripts we must increase our indices by one and thus take N = 11 to obtain ten increments of length $\Delta x = \Delta y = 0.1$. Then, T(1,J), (J=1,2,...,11) corresponds to $v^{2k+1}_{o,j}$, (j=0,1,...,10), and T(I,1), (I=1,2,...,11) corresponds to $v^{2k+1}_{i,o}$, (i=0,1,...,10), at time $(2k+1)\Delta t$.

Using this notation, and the approximations (16) and (17), the difference equations at the boundaries x = 0 and y = 0 are

$$\frac{T^*(1,J) - T(1,J)}{\Delta t/2} = \frac{2T^*(2,J) - 2T^*(1,J)}{(\Delta x)^2} + \frac{T(1,J-1)-2T(1,J)+T(1,J+1)}{(\Delta y)^2}$$

or

$$(11b)\quad 2(\varphi+1)T^*(1,J)-2T^*(2,J) = T(1,J-1)+2(\varphi-1)T(1,J)+T(1,J+1) \quad \text{for } (J=2,3,\ldots,N-1)$$

$$2(\varphi+1)T^*(1,1)-2T^*(2,1) = 2T(1,2)+2(\varphi-1)T(1,1) \quad \text{for } J = 1,$$

and

$$\frac{T(I,1)-T^*(I,1)}{\Delta t/2} = \frac{T^*(I-1,1)-2T^*(I,1)+T^*(I+1,1)}{(\Delta x)^2} + \frac{2T(I,2)-2T(I,1)}{(\Delta y)^2},$$

or

$$(12b)\quad 2(\varphi+1)T(I,1)-2T(I,2) = T^*(I-1,1)+2(\varphi-1)T^*(I,1)+T^*(I+1,1) \quad \text{for } (I=2,3,\ldots,N-1)$$

$$2(\varphi+1)T(1,1) = 2T(1,2) = 2T^*(2,1) + 2(\varphi-1)\,T^*(1,1) \quad \text{for } I = 1$$

Now using equations (11a), (11b), (12a), and (12b) we can write out the systems of simultaneous equations which are to be solved. Theses are:

Implicit in the x direction

$$\begin{aligned}
2(\varphi+1)T^*(1,J) - 2T^*(2,J) &= D(1)\\
-T^*(1,J)+2(\varphi+1)T^*(2,J)-T^*(3,J) &= D(2)\\
-T^*(I-1,J)+2(\varphi+1)T^*(I,J)-T^*(I+1,J) &= D(I)\\
-T^*(N-3,J)+2(\varphi+1)T^*(N-2,J)-T^*(N-1,J) &= D(N-2)\\
-T^*(N-2,J)+2(\varphi+1)T^*(N-1,J) &= D(N-1)
\end{aligned}$$

where $D(I) = T(I,J-1) + 2(\varphi-1)T(I,J) + T(I,J+1)$
$(I=1,2,\ldots,N-1) \quad (J=2,3,\ldots,N-1)$,

and $D(I) = 2T(I,2) + 2(\varphi-1)\,T(I,1)$
$(I=1,2,\ldots,N-1) \qquad J=1$

Implicit in the y direction

$$2(\varphi+1)T(I,1) - 2T(I,2) = D(1)$$
$$-T(I,1) + 2(\varphi+1)T(I,2) - T(I,3) = D(2)$$
$$-T(I,J-1) + 2(\varphi+1)T(I,J) - T(I,J+1) = D(J)$$
$$-T(I,N-3) + 2(\varphi+1)T(I,N-2) - T(I,N-1) = D(N-2)$$
$$-T(I,N-2) + 2(\varphi+1)T(I,N-1) = D(N-1),$$

where $D(J) = T^*(I-1,J) + 2(\varphi-1)T^*(I,J) + T^*(I+1,J)$

$(J = 1,2,\ldots,N-1) \qquad (I = 2,3,\ldots,N-1)$

and $D(J) = 2T^*(2,J) + 2(\varphi-1)T^*(1,J)$

$(J = 1,2,\ldots,N-1) \qquad I=1.$

If we call the coefficients of the main diagonal terms B(I) and the coefficients of the lower diagonal and upper diagonal terms A(I) and C(I) respectively, then

$A(I) = -1$, $(I = 2,3,\ldots,N-1)$ $C(I) = -1$, $(I = 1,2,\ldots,N-2)$

$B(I) = 2(\varphi+1)$, $(I = 1,2,\ldots,N-1)$. In addition, let $F = 2(\varphi-1)$.

The algorithm for solving these systems is then

$$\beta(1) = B(1), \qquad \gamma(1) = D(1)/\beta(1)$$

$$\beta(I) = B(I) - A(I)C(I-1)/\beta(I-1), \qquad \gamma(I) = [D(I) - A(I)\gamma(I-1)]/\beta(I)$$
$$(I = 2,3,\ldots,N-1)$$

$$T^*(N-1) = \gamma(N-1), \qquad T^*(I) = \gamma(I) - C(I)T^*(I+1)/\beta(I)$$

$$T(N-1) = \gamma(N-1), \qquad T(I) = \gamma(I) - C(I)T(I+1)/\beta(I)$$
$$(I = N-2,N-3,\ldots,1)$$

The flow charts for the solution program (IADMT) and the algorithm subroutine (TDIAG) are given in Figures 1 and 2. Twelve time-steps, Δt = 0.001, 0.001, 0.001, 0.002, 0.005, 0.01, 0.01, 0.02, 0.05, 0.05, 0.05, and 0.05, were used consecutively to obtain solutions at times 0.001, 0.002, 0.003, 0.005, 0.01, 0.02, 0.03, 0.05, 0.10, 0.15, 0.20, and 0.25.

The analytic solution for the problem was obtained by the separation of variables technique, and is

$$U(x,y,t) = (16/\pi^2)\sum_{n=1}^{\infty}\sum_{m=1}^{\infty}\frac{(-1)^{n+m}}{(2n-1)(2m-1)}e^{-[(2n-1)^2+(2m-1)^2]\frac{\pi^2}{4}t}\cos(2n-1)\frac{\pi x}{2}\cos(2m-1)\frac{\pi y}{2}$$

A program was written to obtain the analytic solution at the grid points ($i\,\Delta x$, $j\,\Delta y$), ($i,j = 1,2,\ldots,N-1$), for the times 0.005, 0.01, 0.05, and 0.250. This solution is called ANSOL(I,J), and includes all terms in the double summation for n,m = 1 to 20. The values of ANSOL(I,J) were read into the program IADMT for comparison with the values of the numerical solution, TEND(I,J). The difference, W(I,J) = ANSOL(I,J) - TEND(I,J), and the percent difference, Z(I,J) = 100 W(I,J)/ANSOL(I,J), were calculated for all points with the exception of those points on the boundaries where the temperature is held at zero.

Figure 3 shows a print-out of the numerical solution and analytic solution for time 0.25. Figure 4 shows the difference and percent difference for the same time.

Numerical solutions were obtained for several values of Δt and Δx, all at time 0.005. Table 1 lists the values of the numerical solution $v(x,y,t)$ and the percent difference Z(I,J) at the point (0.6,0.6). The analytic solution for this point is $U(x,y,t)$ = 0.9999.

Table 1. Numerical Solution and Percent Difference at Point (0.6,0.6) for Different Values of Δt and Δx.

Δt	Δx	$(\Delta x)^2$	$\Delta t + (\Delta x)^2$	v(x,y,t)	Z(I,J)
0.0005	0.025	0.00062	0.00112	0.9996	0.033
0.001	0.025	0.00062	0.00162	0.9995	0.044
0.0025	0.025	0.00062	0.00312	0.9991	0.088
0.005	0.025	0.00062	0.00562	0.9981	0.180
0.0005	0.05	0.0025	0.0030	0.9994	0.055
0.001	0.05	0.0025	0.0035	0.9994	0.055
0.0025	0.05	0.0025	0.0050	0.9990	0.099
0.005	0.05	0.0025	0.0075	0.9982	0.170
0.0005	0.1	0.01	0.0105	0.9975	0.240
0.001	0.1	0.01	0.0110	0.9975	0.240
0.0025	0.1	0.01	0.0125	0.9973	0.260
0.005	0.1	0.01	0.0150	0.9965	0.340

All solution programs were run on the IBM 360 computer. This computer uses seven digits to the right of the decimal point in a decimal number. Therefore, we can write as a bound for the round-off error $C \times 10^{-6}$, where C is a positive constant. (See McCracken and Dorn [10] .) C is determined by the number of arithmetic operations performed and by how the round-off accumulates during these operations. It is therefore possible for C to become quite large. However, the satisfactory results obtained in all solutions lead us to believe that random cancellation of round-off has occurred, thus keeping C small. Furthermore, since the discretization error depends on Δt and $(\Delta x)^2$, and the smallest of these values was 6.2×10^{-4}, we shall neglect the round-off error and examine only the discretization error.

From Douglas [9] , the discretization error is bounded by the quantity $C_1(\Delta t) + C_2(\Delta x)^2$, where C_1 and C_2 are positive constants and depend on the least upper bounds of the derivatives U_{tt} and U_{xxxx}. We can write this bound as $K\left[\Delta t + (\Delta x)^2\right]$, where $K = \max C_1, C_2$. Thus we say that the discretization error is of $0\left[\Delta t + (\Delta x)^2\right]$.

For convenience, we have used $\Delta t \times 10^3$ and $(\Delta x)^2 \times 10^3$ when working with the data from Table 1. This data was used to plot percent error versus $[\Delta t + (\Delta x)^2] \times 10^3$; the result is shown in Figure 5. The line shown in this figure was obtained by a least squares fit. Although the data appears to fit a straight line, we can not take this as verification that the discretization error is, in fact, of $0[\Delta t + (\Delta x)^2]$. To obtain verification we would need to know, or at least have some estimate of, the derivatives U_{xxxx} and U_{tt}. Knowing the analytic solution, it would be possible to calculate these derivatives; this was not done since the objective of the paper is to illustrate the use of the IAD method rather than to verify predicted results.

We now compare the work required using the IAD method against that required when using the explicit method. We make this comparison for the solution at time 0.25. Although the IAD method is stable for any size time-step, we have seen that the discretization error depends on Δt. Therefore, we avoid using very large time-steps. For this problem, 0.25 was the final value of t, and solutions were obtained for 11 smaller values of t by using values of Δt ranging from 0.001 to 0.05. To find the number of arithmetic operations required for the IAD method in this problem, we take as our starting point the solution of equation (11a), which gives the temperatures implicit in the x direction. As we have seen, this equation results in N-1 = 10 equations in N-1 unknowns. The algorithm used to solve these equations involves three expressions, each of which required three arithmetic operations (including addition and subtraction). These expressions were formed a total of N-2 times, so that

$9(N-2)$ arithmetic operations are required for the solution of each system of $N-1$ equations. To obtain the temperature at every grid point at the end of a half time-step, equation (11a) must be used $N-1$ times. Thus for a half time-step, $9(N-2)\cdot(N-1)$ operations are required. Now, the solution of equation (12a), which gives the temperatures implicit in the y direction, requires the same number of operations for a half time-step. Thus the solution for a complete time-step requires $2x9(N-2)\cdot(N-1)$ operations. As stated, a solution at time 0.25 was obtained as the twelfth complete time-step, so that the total number of arithmetic operations required using the IAD method was

$$12x2x9(N-2)\cdot(N-1) = 19,440$$

Use of the explicit equation (9) requires that $\Delta t \leq (\Delta x)^2/4 = 0.0025$. This means that the solution must be stepped through 100 time-steps to reach time 0.25. The solution of equation (9) involves 7 arithmetic operations, and this equation must be solved at each of the $(N-1)^2$ grid points. A total of $100x7(N-1)^2 = 70,000$ operations are required for a solution at time 0.25. Thus we see that the IAD method requires two-thirds less work than the explicit method for this problem. This problem involved 100 grid points. As the number of grid points increases, it becomes even more advantageous to use the IAD method rather than the explicit method. Note also that the maximum time-step used for this problem was 0.05, which is 20 times larger than the maximum allowable time-step for the explicit method.

Example 2. Solution of a Steady State Problem Using the IAD Method.

Problem: To find the steady state temperature distribution at the face of an infinite prism with edge-length one.

The temperature distribution for this infinite prism (See Figure 26) is governed by the elliptic partial differential equation

$$U_{xx}(x,y) + U_{yy}(x,y) = 0,$$

with boundary conditions

$$U(0,y) = U(x,0) = 0,$$

$$U(x,1) = 1$$

$$U_x(1,y) = -hU(1,y).$$

The last boundary condition means that the boundary $x = 1$ is a partially-conducting boundary. For convenience we take the value of the constant h to be one.

In solving this problem by the IAD method, we treat it as the limiting case of the time-dependent problem $U_{xx} + U_{yy} = U_t$. This allows us to use the same difference equations, (11a) and (12a), that we used in Example 1.

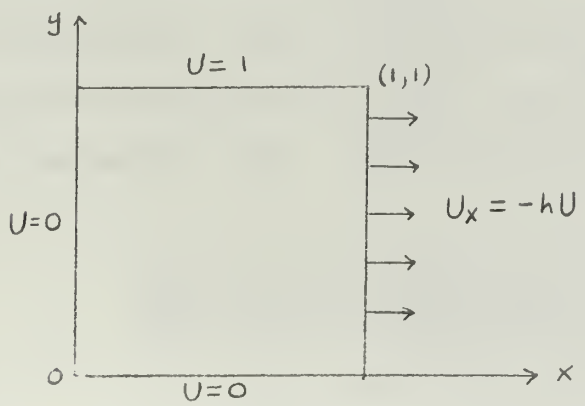

Figure 26. Temperatures at the Face of an Infinite Prism.

We divide the region shown in Figure 26 in exactly the same way in which we divided the region for Example 1. Thus the resulting grid is identical to the grid shown in Figure 25.

THE IAD method, as applied to this steady state problem, can be considered as a process of iteration. In such a process we use a starting approximation to the solution, and then attempt to improve this approximation. We call each attempt at improvement an iteration. Thus we regard the use of both equations (11a) and (12a) as one complete iteration, and we regard the quantity φ appearing in these equations as an iteration parameter. We now wish to determine the optimum values of φ, i.e., the value or values of φ which will give us the most improvement in our starting approximation after the smallest number of iterations.

Recall from the discussion of the convergence of the IAD method that, for a rectangular region, it is possible to predict optimum values of φ. In that discussion, the effect of one complete iteration was represented by

$$E_k = P(\varphi_k)E_{k-1},$$

where $P(\varphi_k)$ is a matrix operator and E_k is the difference between the true solution, $U(x,y)$, and the numerical solution, $v_k(x,y)$, obtained after the kth iteration. We call E_k the error at the kth iteration. We saw previously that we were assured convergence (for an infinite number of iterations) if the eigenvalues of the operator P were all less than one in absolute value. These eigenvalues are given by,

(15a)
$$\frac{(\varphi - 4\sin^2 p\frac{\Delta x}{2\pi})(\varphi - 4\sin^2 q\frac{\Delta y}{2\pi})}{(\varphi + 4\sin^2 p\frac{\Delta x}{2\pi})(\varphi + 4\sin^2 q\frac{\Delta y}{2\pi})}$$

(The difference between this expression and (15) is due to the fact that the region for this problem has side-length one).

From (15a) we see that each of the N-1 eigenvalues of P can be reduced to zero on some one of N-1 iterations if we use values of φ given by

(18) $\varphi_p = 4 \sin^2 p\pi/2N, \quad (p = 1,2,\ldots,N-1).$

Thus, theoretically we should be able to reduce the error to zero after N-1 = 10 iterations. Of course, complete error reduction in 10 iterations could happen only when using an exact numerical procedure. Due to round-off error we do not have an exact procedure, so we can only hope that error will be reduced to a very small, and, therefore, acceptable level. Moreover, since in the IAD method, one complete iteration involves the solution of both equations (11a) and (12a), we require 20 intermediate iterations for maximum error reduction.

Using equation (18), the values of φ and the associated values of Δt (from $\Delta t = (\Delta x)^2/\varphi$) were calculated and are listed in Table 2.

Table 2. Calculated values of the iteration parameter φ_p

p	φ_p	Δt
1	0.02462	0.40612
2	0.21799	0.04587
3	0.58579	0.01707
4	1.09202	0.00916
5	1.68713	0.00593
6	2.31287	0.00432
7	2.90798	0.00344
8	3.41421	0.00293
9	3.78201	0.00264
10	3.97538	0.00252

Since this is a steady state problem, Δt has no real meaning. We use it here only to emphasize the similarity in the time-dependent and steady state problems when using the IAD method. As in the time-dependent problem, where we obtained solutions in order of increasing Δt, we also use the values of φ given in Table 2 in order of increasing Δt, i.e., we use the larger values of φ first. Each

value of φ must be used over a complete iteration so that any particular φ is used for two intermediate iterations.

It is possible to reduce the number of iterations by grouping values of φ. From Table 2 it can be seen that for larger p the difference between values of φ is small. Therefore, an average value of φ may be used for certain groups of φ_p. The eigenvalues of P corresponding to the values of p falling within that group, while not zero, are sufficiently small so that effective error reduction still occurs. For reasonably small N (as in this case), it is possible to calculate all values of φ and set up the groups by inspection. A procedure for setting up groups when N is large is given in [7] . For this problem we group φ as in Table 3.

Table 3. Grouped Values of φ.

p	avg Δt	$\varphi = (\Delta x)^2/\Delta t$
8,9,10	0.00270	3.7037
5,6,7	0.00455	2.1978
3,4	0.01310	0.7633
2	0.04587	0.2179
1	0.40612	0.0246

Thus, by grouping values of φ, we are able to reduce the number of iterations required by one-half with the expectation that the results obtained should be nearly as good as those obtained when using all calculated values of φ. Douglas and Peaceman [11] give examples in which simple values of φ are used. They start with a value between 4 and 10, and then halve this value repeatedly to obtain additional values of φ. We shall obtain solutions to this problem using all three approaches.

For a starting approximation, corresponding to the initial condition in the unsteady state problem, we use

(19) $U(x,y) = xy/(1 - y + x)$.

This expression satisfies the boundary conditions at all but the conducting boundary.

As in Example 1, we can use equations (11a) and (12a) at all interior grid points and at the boundaries where the values of $U(x,y)$ are known. At the conducting boundary, we may use the same approximations for U_t and U_{yy} but must find a new approximation for U_{xx}. Again we assume a fictitious point, a distance Δx outside the boundary, at which the temperature is $U(x+\Delta x,y)$, and as before we write the Taylor expansions for $U(x+\Delta x,y)$ and $U(x-\Delta x,y)$ in terms of temperatures $U(x,y)$ on the boundary, where (x,y) represents the point $[N\Delta x, j\Delta y]$. Combining these expressions, we have, as before

(20) $U(x-\Delta x,y) + U(x+\Delta x,y) = 2U(x,y) + (\Delta x)^2 U_{xx}(x,y) + 0(\Delta x)^2$.

However, we no longer have the symmetry which allows us to say that $U(x-\Delta x,y) = U(x+\Delta x,y)$. We must use the boundary condition to evaluate $U(x+\Delta x,y)$. The method here is one due to Schmidt as given in Carslaw and Jaeger [12]. We use a symmetric difference for the normal derivative at the boundary to get

$$U_x(x,y) = 1/(2\Delta x)\left[U(x+\Delta x,y) - U(x-\Delta x,y)\right] + 0(\Delta x)^2$$

Solving for $U(x+\Delta x,y)$, and using the boundary condition

$$U_x(x,y) = -U(x,y)$$

we have

$$U(x+\Delta x,y) = U(x-\Delta x,y) - 2\Delta x U(x,y) + 0(\Delta x)^2.$$

Substituting in (20), we have, finally,

(21) $U_{xx}(x,y) = 2/(\Delta x)^2\left[U(x-\Delta x,y) - (1+\Delta x)U(x,y)\right] + 0(\Delta x)^2$.

In the computer program IADM we use TST(I,J) to denote the approximation to $U(x,y)$ at the end of each intermediate iteration

and TEND(I,J) to denote the approximation at the end of a complete iteration. In the equations that follow we abbreviate TST(I,J) by T*(I,J) and TEND(I,J) by T(I,J). Using this notation and the approximation (21), the difference equations at the conducting boundary are:

Implicit in the x direction

$$\frac{T^*(N,J) - T(N,J)}{\Delta t} = \frac{2T^*(N-1,J) - 2(1+\Delta x)T^*(N,J)}{(\Delta x)^2} + \frac{T(N,J-1) - 2T(N,J) + T(N,J+1)}{(\Delta y)^2}$$

or

$$(11c)\quad -2T^*(N-1,J) + \left[\varphi(k) + 2(1+\Delta x)\right] T^*(N,J) = T(N,J-1) + \left[\varphi(k) - 2\right] T(N,J) + T(N,J+1),$$

where

$$\varphi(k) = (\Delta x)^2/\Delta t_k = (\Delta y)^2/\Delta t_k$$

and k denotes the iteration number.

Implicit in the y direction:

$$\frac{T(N,J) - T^*(N,J)}{\Delta t} = \frac{2T^*(N-1,J) - 2(1+\Delta x)T^*(N,J)}{(\Delta x)^2} + \frac{T(N,J-1) - 2T(N,J) + T(N,J+1),}{(\Delta y)^2}$$

or

$$(12c)\quad -T(N,J-1) + \left[\varphi(k) + 2\right] T(N,J) - T(N,J+1) = 2T^*(N-1,J) + \left[\varphi(k) - 2(1+\Delta x)\right] T^*(N,J).$$

We empahsize here that the value $\varphi(k)$ must be used for a complete iteration before introducing the value $\varphi(k+1)$.

Now, using equations (11a), (12a), (11c), and (12c) we write out the systems of equations to be solved:

Implicit in the x direction

$$[2+\varphi(K)]\,T^*(2,J) - T^*(3,J) = D(2)$$

$$-T^*(2,J)+[2+\varphi(K)]\,T^*(3,J)- T^*(4,J) = D(3)$$

$$-T^*(I-1,J) + [2+\varphi(K)]\,T^*(I,J) - T^*(I+1,J) = D(I)$$

$$-T^*(N-2,J) +[2+\varphi(K)]\,T^*(N-1,J) - T^*(N,J) = D(N-1)$$

$$-T^*(N-1,J) + [\varphi(K) +2(1+\Delta x)]\,T^*(N,J) = D(N)$$

$(J=2,3,\ldots,N-1)$

where $D(I) = T(I,J-1) + [\varphi(k) - 2]\,T(I,J) + T(I,J+1)$

$(I=2,3,\ldots,N)$ $(J=2,3,\ldots,N-1)$

As in Example 1, the coefficients are A(I), B(I) and C(I), with

$B(I) = 2 + \varphi(k)$, $(I=2,3,\ldots,N-1)$, $B(N) = \varphi(k) + 2(1+\Delta x)$

$A(I) = -1$, $(I=3,4,\ldots,N-1)$, $A(N) = -2$

$C(I) = -1$, $(I = 2,3,\ldots,N-1)$

Implicit in the y direction

$$[2+\varphi(K)]\,T(I,2) - T(I,3) = D(2)$$

$$-T(I,2)+[2+\varphi(K)]\,T(I,3) - T(I,4) = D(3)$$

$$-T(I,J-1)+[2+\varphi(K)]\,T(I,J) - T(I,J+1) = D(J)$$

$$-T(I,N-3)+[2+\varphi(K)]\,T(I,N-2) - T(I,N-1) = D(N-2)$$

$$-T(I,N-2) + [2+\varphi(K)]\,T(I,N-1) = D(N-1)$$

$(I=2,3,\ldots,N)$

where $D(J) = T^*(I-1,J) + [\varphi(k) - 2]\,T^*(I,J) + T^*(I+1,J)$

$(I=2,3,\ldots,N-1)$ $(J-2,3,\ldots,N-2)$

$D(N-1) = T^*(I-1,N-1) + [\varphi(k) - 2]\,T^*(I,N-1) + T^*(I+1,N-1)$

$+ T(I,N)$

$(I=2,3,\ldots,N-1)$

and $D(J) = 2T^*(N-1,J) + [\varphi(k) - 2(1+\Delta x)]\,T^*(N,J)$

$I=N$, $(J=2,3,\ldots,N-2)$

$$D(N-1) = 2T^*(N-1,N-1) + \left[\varphi(k) - 2(1+\Delta x)\right] T^*(N,N-1) + T(N,N).$$

We note that D(N-1) contains the term T(I,N) which has been transferred from the left-hand side since it is a known quantity, by virtue of the boundary condition U(x,1) = 1.

The coefficients are written

$$B(I) = 2 + \varphi(k), \quad (I=2,3,\ldots,N-1)$$

$$A(I) = -1, \quad (I=3,4,\ldots,N-1)$$

$$C(I) = -1, \quad (I=2,3,\ldots,N-2) \quad .$$

We use the algorithm given in Example 1 to solve these two systems of equations. The flow charts for the solution program IADM and the algorithm subroutine TDIAG are given in Figures 6 and 2.

The analytic solution to this problem was obtained from Churchill [13] and is

$$(22) \quad U(x,y) = 2\sum_{n=1}^{\infty} \frac{A_n \sinh \alpha_n y}{\alpha_n \sinh \alpha_n} \sin \alpha_n x$$

where $$A_n = \frac{1 - \cos \alpha_n}{1 + \cos^2 \alpha_n}$$

and $\alpha_1, \alpha_2, \ldots$ are the roots of $\tan \alpha = -\alpha$.

A program was written to obtain the analytic solution at the grid points $(i\,\Delta x, j\,\Delta y)$, (i,j=2,3,...,N), and this solution is called V(I,J). To obtain V(I,J), the series in (22) was summed until a term was reached for which the absolute value of that term, divided by the sum of previous terms, was less than 10^{-6}. The summation was terminated upon reaching such a term. The values of V(I,J) were read into the program IADM to allow comparison with the numerical solution, TEND(I,J). IADM calculates the difference,

W(I,J) = V(I,J) - TEND(I,J) and the percent difference,

Z(I,J) = 100 W(I,J)/V(I,J) for all grid points except at the known boundaries, x = y = 0 and y = 1.

The first solution was obtained using the five grouped values of φ listed in Table 3. Figures 7 through 16 give the results for all five iterations. In these figures as well as in thosewhich follow, the odd numbered figures show the numerical and analytic solutions for a particular iteration while the even numbered figures show the difference and percent difference. Results of all five iterations are shown, so that the effect of each value of φ and the process of error reduction can be seen. We note that error reduction takes place diagonally from the larger grid points to the smaller. This is because the larger values of φ which are used first correspond to the larger eigenvalues of the operator P, and these eigenvalues in turn correspond to larger values of x and y.

The second solution was obtained using the ten values of φ listed in Table 2. Results of the tenth iteration are shown in Figures 17 and 18. Comparison of Figures 16 and 18 shows that the results of using the grouped values of φ in five iterations are essentially the same as the results of using all calculated values of φ in ten iterations. In both cases errors range from approximately 0.05 percent to 2.5 percent. We therefore conclude that by grouping values of φ we can reduce the number of required iterations significantly.

The third solution was obtained using the five simple values of φ: 4,2,1,0.5, and 0.25. Results are shown in Figures 19 and 20. It can be seen that error reduction was less than in the solution obtained by using five grouped values of φ. However, these results

are considered to be satisfactory when the resulting simplification of the problem is taken into account.

We use the results of the second solution (Figure 16) to make a comparison of the work requirement for the IAD method and the work requirement for another iteration method. In the IAD method, the equation $U(x,y) = xy/(1 - y + x)$ was used to obtain starting values at each grid point. Comparison of these values with the analytic solution shows a maximum difference (maximum initial error) of approximately 0.10. Comparison of the numerical solution for the last iteration and the analytic solution shows a maximum difference (maximum final error) of 0.0041. Thus the reduction of error is on the order of 4×10^{-2}. In Example 1 we found that a complete time-step using the IAD method required $2 \times 9(N-2) \cdot (N-1)$ arithmetic operations (including addition and subtraction). For this problem a complete time-step corresponds to a complete iteration, and since five complete iterations were used, the total number of operations required for an error reduction of 4×10^{-2} was $5 \times 2 \times 9(N-2) \cdot (N-1) = 8100$. Peaceman and Rachford [7] give the work requirement for the extrapolated Liebmann iteration method of solution to Laplace's equation as $5\nu(N-1)^3$ for an error reduction of $10^{-\nu}$. Thus, for an error reduction of 10^{-2}, the number of arithmetic operations by the extrapolated Liebmann method is $5 \times 2(N-1)^3 = 10{,}000$. Thus the IAD method requires fewer operations for a larger error reduction than the extrapolated Liebmann method, considered to be one of the more efficient methods of iteration.

Example 3. Solution of a Time-Dependent Problem in a Non-rectangular Region Using the IAD Method.

Problem: To find the temperature with time for a quarter section of the face of an infinite cylinder with radius one.

The temperature at the face of this infinite cylinder (See Figure 27) is governed by the following parabolic partial differential equation

$$(23) \quad u_{rr} + \frac{1}{r} u_r + \frac{1}{r^2} u_{\theta\theta} = \frac{1}{k} u_t$$

with boundary and initial conditions

$$u(r,0,t) = u(r, \pi/2, t) = 1, \quad u(1,\theta,t) = 1 \text{ and } u(r,\theta,0) = 0$$

As before, we choose the time unit such that the diffusivity k is one.

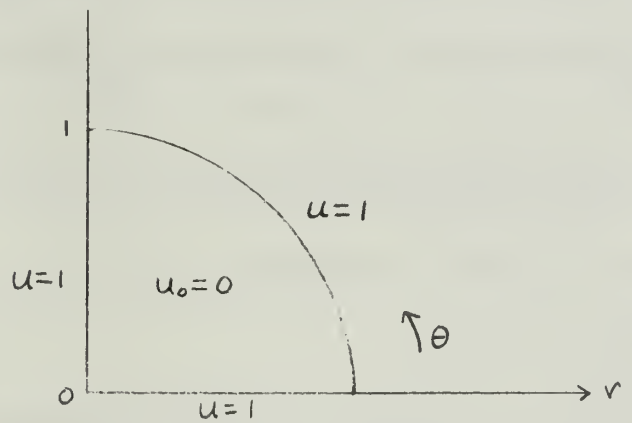

Figure 27. Temperatures at the Face of an Infinite Cylinder.

We divide the region shown in Figure 27 into N increments of length $\Delta r = 1/N$ in the r direction, and into N angles of magnitude $\Delta\theta = \pi/2N$ in the θ direction. The resulting grid is shown in Figure 28.

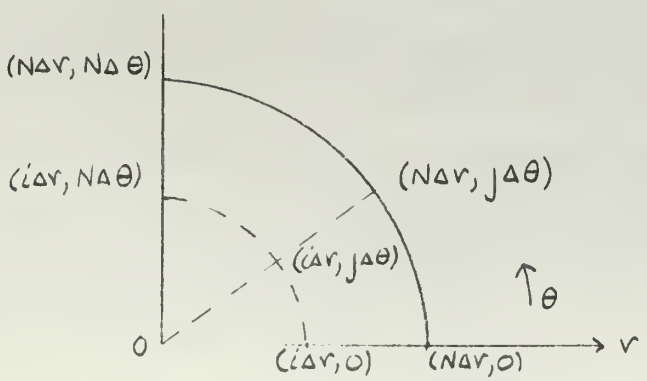

Figure 28. Grid for the Region Shown in Figure 27.

Let $v(r,\theta,t)$ be the difference approximation to $u(r,\theta,t)$. We then use the following difference approximations for the partial derivatives u_t, u_r, u_{rr}, and $u_{\theta\theta}$:

$$V_t = \frac{V(r,\theta,t+\Delta t) - V(r,\theta,t)}{\Delta t/2} + O(\Delta t)$$

$$V_r = \frac{V(r+\Delta r,\theta,t) - V(r-\Delta r,\theta,t)}{2\Delta r} + O(\Delta r)^2$$

$$V_{rr} = \frac{V(r-\Delta r,\theta,t) - 2V(r,\theta,t) + V(r+\Delta r,\theta,t)}{(\Delta r)^2} + O(\Delta r)^2$$

$$V_{\theta\theta} = \frac{V(r,\theta-\Delta\theta,t) - 2V(r,\theta,t) + V(r,\theta+\Delta\theta,t)}{(\Delta\theta)^2} + O(\Delta\theta)^2$$

Now consider the two terms $u_{rr} + (1/r)u_r$, of (23). They are approximated by

$$\frac{V(r+\Delta r,\theta,t) - V(r-\Delta r,\theta,t)}{2r\Delta r} + \frac{V(r-\Delta r,\theta,t) - 2V(r,\theta,t) + V(r+\Delta r,\theta,t)}{(\Delta r)^2}$$

$$= \frac{(1-\frac{\Delta r}{2r})V(r-\Delta r,\theta,t) - 2V(r,\theta,t) + (1+\frac{\Delta r}{2r})V(r+\Delta r,\theta,t)}{(\Delta r)^2} + O(\Delta r)^2$$

The complete difference equation approximating (23) is

$$(24)\quad \frac{V(r,\theta,t+\Delta t) - V(r,\theta,t)}{\Delta t/2}$$
$$-\frac{\left[(1-\frac{\Delta r}{2r})V(r-\Delta r,\theta,t) - 2V(r,\theta,t) + (1+\frac{\Delta r}{2r})V(r+\Delta r,\theta,t)\right]}{(\Delta r)^2}$$
$$-\frac{(\Delta r)^2}{r^2(\Delta\theta)^2}\left[\frac{V(r,\theta-\Delta\theta,t) - 2V(r,\theta,t) + V(r,\theta+\Delta\theta,t)}{(\Delta r)^2}\right]$$
$$-\left[O(\Delta t) + O(\Delta r)^2 + O(\Delta\theta)^2\right] = 0.$$

Now consider (24) as an operator $L_r(v) = 0$. If L_r operates on $u(r,\theta,t)$, we have

$$L_r(u) = \frac{u(r,\theta,t+\Delta t) - u(r,\theta,t)}{\Delta t/2} + \frac{\Delta t}{2}\tilde{u}_{tt} + O(\Delta t)^2$$
$$-\frac{\left[(1-\frac{\Delta r}{2r})u(r-\Delta r,\theta,t) - 2u(r,\theta,t) + (1+\frac{\Delta r}{2r})u(r+\Delta r,\theta,t)\right]}{(\Delta r)^2}$$
$$-\frac{(1+2r)}{24r}(\Delta r)^2\tilde{u}_{rrrr} + O(\Delta r)^4 - \frac{u(r,\theta-\Delta\theta,t) - 2u(r,\theta,t) + u(r,\theta+\Delta\theta,t)}{r^2(\Delta\theta)^2}$$
$$-\frac{(\Delta\theta)^2}{12r^2}\tilde{u}_{\theta\theta\theta\theta} + O(\Delta\theta)^4 = 0$$

where the tilde indicates that the derivatives are to be taken at certain points in the region $0 < r < 1$, $0 < \theta < \pi/2$, $0 \leq t \leq T$.

We call $z(r,\theta,t) = v(r,\theta,t) - u(r,\theta,t)$ the discretization error. Then since L_r is a linear operator, we have

$$L_r(v) - L_r(u) = L_r(v - u),$$

or

$$L_r(z) = -\frac{\Delta t}{2}\tilde{u}_{tt} + \frac{1+2r}{24r}(\Delta r)^2\tilde{u}_{rrrr} + \frac{(\Delta\theta)^2}{12r^2}\tilde{u}_{\theta\theta\theta\theta} + O(\Delta t)^2 + O(\Delta r)^4 + O(\Delta\theta)^4$$

Call the right hand side of this equation $w(r,\theta,t)$. Thus we see that $z(r,\theta,t)$ is a solution to the difference equation

$$L_r(z) = w(r,\theta,t)$$

Since (24) represents a convergent difference scheme, we have that

$$\text{l.u.b } |z| \leq \text{l.u.b } |w|,$$

or

$$(25) \quad |z| \leq C_1\Delta t + C_2(\Delta r)^2 + C_3(\Delta\theta)^2,$$

where C_1, C_2 and C_3 depend on the least upper bounds of u_{tt}, u_{rrrr} and $u_{\theta\theta\theta\theta}$ respectively. In addition, C_2 and C_3 depend on r. Therefore, we should expect a discretization error of $O\left[\Delta t + (\Delta r)^2 + (\Delta\theta)^2\right]$.

We now digress somewhat to consider the IAD method as applied to the related elliptic PDE $(u_{rr} + (1/r)u_r + (1/r^2)u_{\theta\theta} = 0)$ for this non-rectangular region. If $U(r,\theta)$ represents a vector solution to this equation, then the difference analog to (24) is

$$-\frac{1}{(\Delta r)^2}\left[A^h + A^v\right] \; U(r,\theta) = 0$$

where A^h and A^v are square matrices such that

$$A^h U(r,\theta) = -\left(1 - \frac{\Delta r}{2r}\right) U(r-\Delta r,\theta) + 2U(r,\theta) - \left(1 + \frac{\Delta r}{2r}\right) U(r+\Delta r,\theta)$$

$$A^v U(r,0) = -\frac{(\Delta r)^2}{r^2(\Delta\theta)^2}\left[U(r,\theta-\Delta\theta) - 2U(r,\theta) + U(r,\theta+\Delta\theta)\right] .$$

For the simple case of N = 4 (a net with 9 interior points) we find that

$$A^h = \begin{bmatrix} 2 & -3/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3/4 & 2 & -5/4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -5/6 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -3/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3/4 & 2 & -5/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5/6 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -3/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3/4 & 2 & -5/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5/6 & 2 \end{bmatrix}$$

and A^v is similar to a matrix B such that

$$B = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1/4 & 1/2 & -1/4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/9 & 2/9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/4 & 1/2 & -1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/9 & 2/9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/4 & 1/2 & -1/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/9 & 2/9 \end{bmatrix}$$

For this case, although A^h and A^v are non-singular, neither is symmetric, and a calculation shows that $A^h A^v \neq A^v A^h$. Thus, for this non-rectangular region, A^h and A^v do not share a common eigenvector basis, and it would therefore not be possible by the method used in Example 2 to predict values of the iteration parameter for optimum convergence rate. It would be necessary to use simple values for this iteration parameter (See Example 2). This is considered to be the most serious disadvantage of the IAD method. However, for the parabolic problem under consideration, we shall see that entirely satisfactory results are possible even though the region is non-rectangular.

We now return to equation (24). Call $V^k_{i,j}$ the approximation to $u(r,\theta,t)$ at the grid point $(i\Delta r,\ j\Delta\theta,\ k\Delta t)$. Let $\varphi = (\Delta r)^2/\Delta t$ and $\alpha = \Delta\theta$; also note that $r = i\Delta r$. Then equation (24) yields the "working equations" analogous to equations (11a) and (12a). These are:

Implicit in the r direction

$$(26)\quad -\left(1-\frac{1}{2i}\right)V^{2K+1}_{i-1,j} + 2(1+\varphi)V^{2K+1}_{i,j} - \left(1+\frac{1}{2i}\right)V^{2K+1}_{i+1,j}$$
$$= \frac{1}{i^2\alpha^2}V^{2K}_{i,j-1} + 2\left(\varphi - \frac{1}{i^2\alpha^2}\right)V^{2K}_{i,j} + \frac{1}{i^2\alpha^2}V^{2K}_{i,j+1}$$

Implicit in the θ direction

$$(27)\quad -\frac{1}{i^2\alpha^2}V^{2K+2}_{i,j-1} + 2\left(\frac{1}{i^2\alpha^2}+\varphi\right)V^{2K+2}_{i,j} - \frac{1}{i^2\alpha^2}V^{2K+2}_{i,j+1}$$
$$= \left(1-\frac{1}{2i}\right)V^{2K+1}_{i-1,j} + 2(\varphi-1)V^{2K+1}_{i,j} + \left(1+\frac{1}{2i}\right)V^{2K+1}_{i+1,j}$$

Since the coefficients in these equations depend on i, this will add some complexity to the computer program. However, since the temperature is given at all boundaries, we need find no special approximations at any boundary, as was necessary in Examples 1 and 2. We use the same notation used in the previous two examples when writing out the two systems of simultaneous equations which result from Equations (26) and (27). These are :

Implicit in the r direction

$$2(\varphi+1)T^*(2,J) - (1+1/2)T^*(3,J) = D(2)$$
$$-(1-1/4)T^*(2,J)+2(\varphi+1)T^*(3,J)-(1+1/4)T^*(4,J) = D(3)$$
$$---$$
$$-(1-1/2(I-1))T^*(I-1,J)+2(\varphi+1)T^*(I,J)-(1+1/2(I-1))T^*(I+1,J)=D(I)$$
$$---$$
$$-1(1-1/2(N-3))T^*(N-3,J)+2(\varphi+1)T^*(N-2,J)-(1+1/2(N-3))T^*(N-1,J)=D(N-2)$$
$$-(1-1/2(N-2))T^*(N-2,J)+2(\varphi+1)T^*(N-1,J) = D(N-1)$$

where

$$D(I) = \frac{1}{(I-1)^2\alpha^2} T(I,J-1) + 2\left(\varphi - \frac{1}{(I-1)^2\alpha^2}\right) T(I,J) + \frac{1}{(I-1)^2\alpha^2} T(I,J+1)$$

$$(J=2,3,\ldots,N-1) \quad (I=2,3,\ldots,N-1)$$

and

$$\bar{D}(2) = D(2) + 0.5; \quad \bar{D}(N-1) = D(N-1) + 1 + 1/(2(N-2)).$$

The coefficients needed for the algorithm subroutine, TDIAG, are

$$B(I) = 2(\varphi+1) \quad (I=2,3,\ldots,N-1)$$

$$A(I) = -(1-1/2(I-1)) \quad (I=3,4,\ldots,N-1)$$

$$C(I) = -(1+1/2(I-1)) \quad (I=2,3,\ldots,N-2).$$

Implicit in the θ direction

$$2\left[\varphi + \frac{1}{(I-1)^2\alpha^2}\right] T(I,2) - \frac{1}{(I-1)^2\alpha^2} T(I,3) = D(2)$$

$$\frac{-1}{(I-1)^2\alpha^2} T(I,2) + 2\left[\varphi + \frac{1}{(I-1)^2\alpha^2}\right] T(I,3) - \frac{1}{(I-1)^2\alpha^2} T(I,4) = D(3)$$

$$\frac{-1}{(I-1)^2\alpha^2} T(I,J-1) + 2\left[\varphi + \frac{1}{(I-1)^2\alpha^2}\right] T(I,J) - \frac{1}{(I-1)^2\alpha^2} T(I,J+1) = D(J)$$

$$\frac{-1}{(I-1)^2\alpha^2} T(I,N-3) + 2\left[\varphi + \frac{1}{(I-1)^2\alpha^2}\right] T(I,N-2) - \frac{1}{(I-1)^2\alpha^2} T(I,N-1) = D(N-2)$$

$$\frac{-1}{(I-1)^2\alpha^2} T(I,N-2) + 2\left[\varphi + \frac{1}{(I-1)^2\alpha^2}\right] T(I,N-1) = D(N-1),$$

where $D(J) = \left[1 - \frac{1}{2(I-1)}\right] T^*(I-1,J) + 2(\varphi-1) T^*(I,J) + \left[1 + \frac{1}{2(I-1)}\right] T^*(I+1,J)$

$$(I=2,3,\ldots,N-1), \quad (J=2,3,\ldots,N-1),$$

and $\bar{D}(2) = D(2) + \frac{1}{(I-1)^2\alpha^2}$ $\quad \bar{D}(N-1) = D(N-1) + \frac{1}{(I-1)^2\alpha^2}$

with coefficients

$$B(J) = 2(\varphi+1/((I-1)^2\alpha^2)) \quad (J=2,3,\ldots,N-1), (I=2,3,\ldots,N-1)$$

$$A(J) = \frac{-1}{(I-1)^2\alpha^2} \quad (J=3,4,\ldots,N-1), (I=2,3,\ldots,N-1)$$

$$C(J) = A(J) \quad (J=2,3,\ldots,N-2), (I=2,3,\ldots,N-1).$$

Note that in these equations, each coefficient has I reduced by one. This is a consequence of having to avoid zero subscripts.

The flow charts for the solution program, IADMC, and the algorithm subroutine, TDIAG, are given in Figures 21 and 2.

The analytic solution is given in Jaeger [14] for the points $r = 0.25$, 0.5 and 0.75; $\theta = 15^{\circ}$, 30° and 45° (the temperature is symmetric about the 45 degree line) at times between 0.01 and 0.2. These temperatures are given to three significant figures, but the author states that calculations were made to four significant figures so that the third figure is considered to be correct.

Figure 22 shows a printout of the intermediate numerical solution, TST(I,J), and the numerical solution, TEND(I,J), for time 0.1. For this solution we used $\Delta r = 1/12$, $\Delta\theta = \pi/24$ and $\Delta t = 0.005$. Table 4 is a comparison of the analytic and numerical solutions at time 0.1.

Table 4. Analytic and Numerical Solutions for Time 0.1

$r \rightarrow$	0.25		0.5		0.75	
	Anal. Sol.	Num. Sol.	Anal. Sol.	Num. Sol.	Anal. Sol.	Num. Sol.
$\downarrow \theta$						
15°	0.979	0.9788	0.963	0.9631	0.957	0.9574
30°	0.946	0.9453	0.906	0.9053	0.891	0.8907
45°	0.952	0.9518	0.917	0.9166	0.904	0.9037

Nine different solutions were obtained using combinations of Δr, $\Delta\theta = 1/12$, $\pi/24$; $1/24$, $\pi/48$; $1/36$, $\pi/72$ and $\Delta t = 0.005$, 0.0025, 0.00125 for time 0.02. Table 5 shows these results at point $(0.5, 45^{\circ})$. The difference listed in this table is the difference between the numerical solution and the analytic solution at this point. The analytic solution is 0.168.

Table 5. Results of solutions using various Δr, $\Delta\theta$, and Δt at the point $(0.5, 45^{\circ})$

Δt	Δr	$\Delta\theta$	$\Delta t + (\Delta r)^2 + (\Delta\theta)^2$	Num. Soln.	Difference
0.00125	0.0277	0.0436	0.00393	0.167	-0.001
0.0025	0.0277	0.0436	0.00517	0.168	0
0.005	0.0277	0.0436	0.00767	0.171	0.003
0.00125	0.0416	0.0654	0.00728	0.168	0
0.0025	0.0416	0.0654	0.00853	0.169	0.001
0.005	0.0416	0.0654	0.01103	0.172	0.004
0.00125	0.0833	0.1309	0.01345	0.174	0.006
0.0025	0.0833	0.1309	0.01470	0.175	0.007
0.005	0.0833	0.1309	0.01720	0.176	0.008

Figure 23 is a plot of the absolute difference versus $\Delta t + (\Delta r)^2 + (\Delta\theta)^2$ using the data given in Table 5. The line shown in this figure was obtained by least squares and has the equation $0.626x - 2.88$ (we have scaled all values by a factor of 10^3, for convenience). Here, as in Example 1, the data appears to fit the straight line, but, as previously stated, we cannot take this as verification that the discretization error is of $0[\Delta t + (\Delta r)^2 + (\Delta\theta)^2]$ since we do not have a predicted value for K. The analytic solution as given in [14] is

$$u(r,0,t) = 1 - \frac{8}{\pi}\sum_{n=0}^{\infty}\frac{\sin 2(2n+1)\theta}{2n+1}\sum_{m=1}^{\infty} e^{-t\alpha_{s,m}^2}\frac{J_s(r\alpha_{s,m})}{[J_s'(\alpha_{s,m})]^2}\int_0^t J_s(\alpha_{s,m}t)dt$$

where J_s is the Bessel Function of order s, $(s = 2(2n + 1)\theta$, $n = 0,1,\ldots)$, and $\alpha_{s,m}$ $(m = 1,2,\ldots)$ are the positive roots of $J_s(\alpha) = 0$. Thus we see that it would be very difficult to obtain the derivatives u_{tt}, u_{rrrr}, and $u_{\theta\theta\theta\theta}$ needed to predict a value for K.

The complexity of the above equation also leads us to conclude that the IAD method yields a simpler program than one involving the analytic solution.

Another advantage of the IAD method in this example is that, by solving the equation in polar coordinates, we avoid the necessity of interpolation at the circular boundary. Such interpolation becomes necessary when placing a rectangular net on a circular region, since not all of the grid points will fall on the curved boundary.

CONCLUSIONS

From the numerical results of the three examples, we have seen that accurate results are possible when using the IAD method. The results for the parabolic problem in the circular region suggest the applicability of the method to many types of problems, including fluid flow and electric potential. When the differential equation is written in polar coordinates, a possible difficulty arises when dealing with derivative-type boundary conditions at the origin, since the differential equation has a singularity at that point. (Temperatures on the boundary were specified in Example 3, thus avoiding this problem.) Such a boundary condition would require that additional approximations be made. However, the method seems ideally suited to problems involving annular regions. As pointed out previously, one advantage in using the IAD method on equations in polar form is that there is no need for interpolation at curved boundaries. A second advantage is that the computer program for the IAD method is probably simpler to write than a program to obtain temperatures at the grid points using the analytic solution (if known).

Since Example 2 involved an elliptic problem over a rectangular region, we were able to calculate optimal values for the iteration parameter φ. For non-rectangular regions the same procedure is not applicable; we would, therefore, be in doubt as to the number of iterations required. However, it is possible, when using simple values of φ, to gauge the progress by calculating the sum of the squares of the errors after each complete iteration to see what the effect of a particular value of φ might be. Details for this process as well as other examples of solutions in non-rectangular

regions may be found in Douglas and Peaceman [11] . In addition, Wachpress [16] suggests a combination of the IAD method and an iteration method due to Lanczos for elliptic problems in non-rectangular regions. Douglas [15] , [17] also gives details for extension of the IAD method to mildly nonlinear problems and to problems in three space variables.

Some doubt remains whether the data obtained in Examples 1 and 3 actually satisfies the predicted error bounds. As we have seen, the data obtained from the numerical solutions is not sufficient in itself to verify predicted results. The whole question of discretization error is found to be much more complex than was supposed, and, in fact, could be the subject of a separate paper.

The three examples given show that the IAD method involves essentially the same equations for both elliptic and parabolic problems. The algorithm subroutine used to solve the simultaneous systems generated by the IAD equations is exactly the same for both problems and requires only a change of input parameters. This fact, and the satisfactory results obtained, coupled with the ease of application, leads us to conclude that the IAD method has a wide range of application in the field of partial differential equations.

BIBLIOGRAPHY

1. Brice Carnahan, H. A. Luther and James O. Wilkes, Preliminary Edition of Applied Numerical Methods, Volume II, Wiley and Sons, New York, 1964.

2. A. Ralston and H. S. Wilf, Mathematical Methods for Digital Computers, Wiley and Sons, New York, 1960.

3. G. E. Forsythe and W. R. Wasow, Finite-Difference Methods for Partial Differential Equations, Wiley and Sons, New York, 1960.

4. G. G. O'Brien, M. A. Hyman and S. Kaplan, A Study of the Numerical Solution of Partial Differential Equations, Journal of Mathematics and Physics, Volume 29 (1951) pp. 223-251.

5. R. D. Richtmyer, Difference Methods for Initial Value Problems, Interscience Publishers, Inc., New York, 1957.

6. J. Douglas, Jr., On the Relation Between Stability and Convergence in the Numerical Solution of Linear Parabolic and Hyperbolic Differential Equations, J. Soc. Ind. Appl. Math. Vol 4 (1956) pp. 20-37.

7. D. W. Peaceman and H. H. Rachford, The Numercial Solution of Parabolic and Elliptic Differential Equations, J. Soc. Ind. Appl. Math. vol 3 (1955) pp. 28-41.

8. G. Birkhoff and R. S. Varga, Implicit Alternating Direction Methods, Trans. Amer. Math. Soc. Vol 92 (1959) pp. 13-24.

9. J. Douglas, Jr. and H. H. Rachford, On the Numerical Solution of Heat Conduction Problems in Two and Three Space Variables, Trans. Amer. Math. Soc. vol 82 (1956) pp. 421-439.

10. D. D. McCracken and W. S. Dorn, Numerical Methods and Fortran Programming, Wiley and Sons, New York, 1964.

11. J. Douglas, Jr. and D. W. Peaceman, Numerical Solution of Two Dimensional Heat Flow Problems, A. I. Ch. E. Journal vol 1 (1955) pp. 505-512.

12. H. S. Carslaw and J. C. Jaeger, Conduction of Heat in Solids, Oxford University Press, 1959.

13. R. V. Churchill, Modern Operational Mathematics in Engineering, McGraw Hill, New York, 1944.

14. J. C. Jaeger, Heat Conduction in a Wedge, or an Infinite Cylinder Whose Cross-section is a Circle or a Sector of a Circle, Phil. Mag. (7) Vol 33 (1942) pp. 527-536.

15. J. Douglas, Jr., Alternating Direction Methods for Three Space Variables, Numerische Mathematik, vol 4 (1962) pp. 41-63.

16. E. L. Wachspress, Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1966.

17. J. Douglas, Jr., Alternating Direction Iteration for Mildly Nonlinear Elliptic Difference Equations, Numerische Mathematik, vol 3 (1961) pp. 92-98.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library Naval Postgraduate School Monterey, California 93940	2
3.	Office of Naval Research Attn: Dr. Adkins Department of the Navy Washington, D. C. 20350	1
4.	Associate Professor U. R. Kodres Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
5.	LT Frederick J. Leipold, USN U. S. Naval Communications Station San Diego, California 92132	1

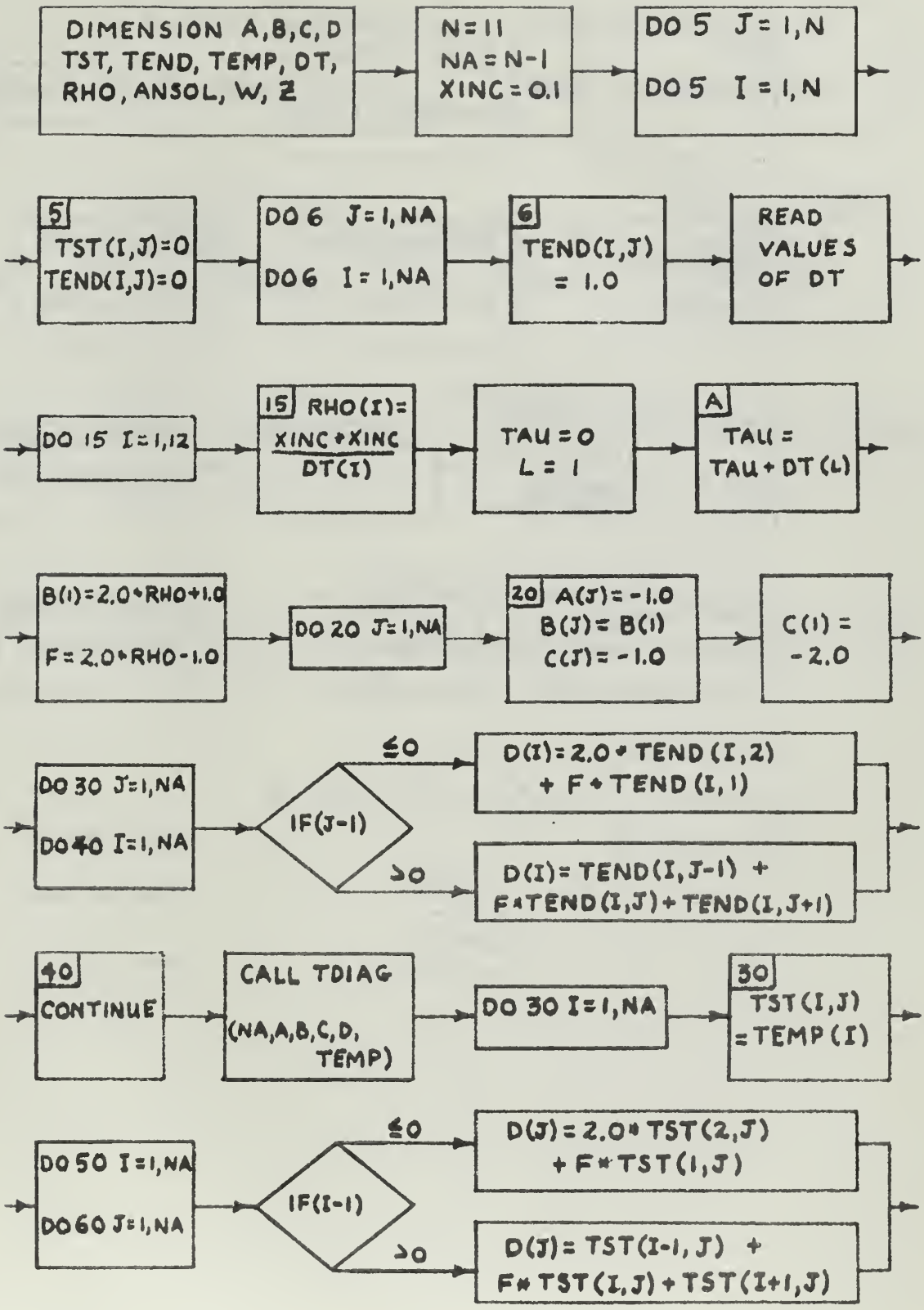

FIGURE 1 FLOWCHART FOR PROGRAM IADMT

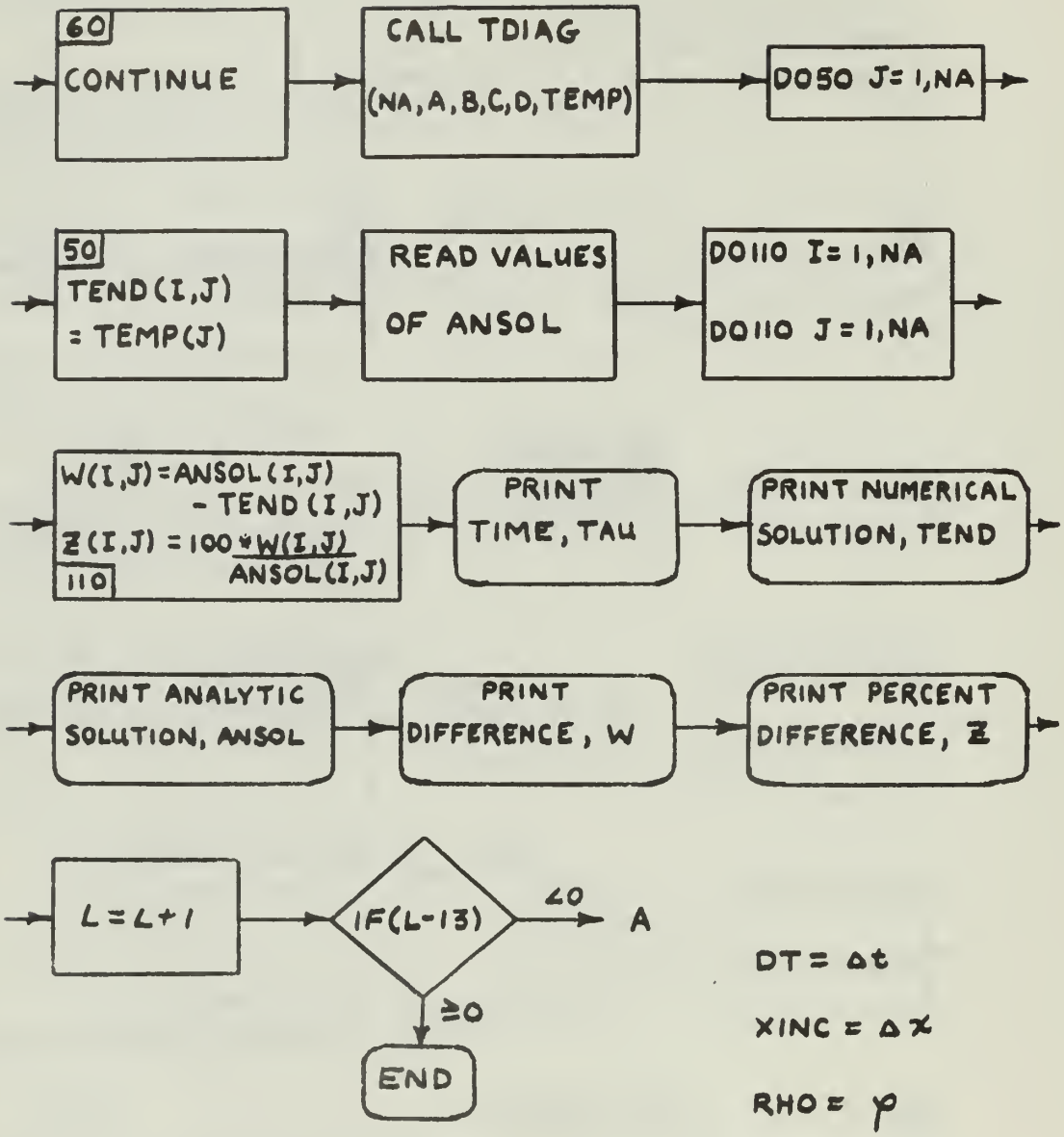

FIGURE 1 (CONTINUED)

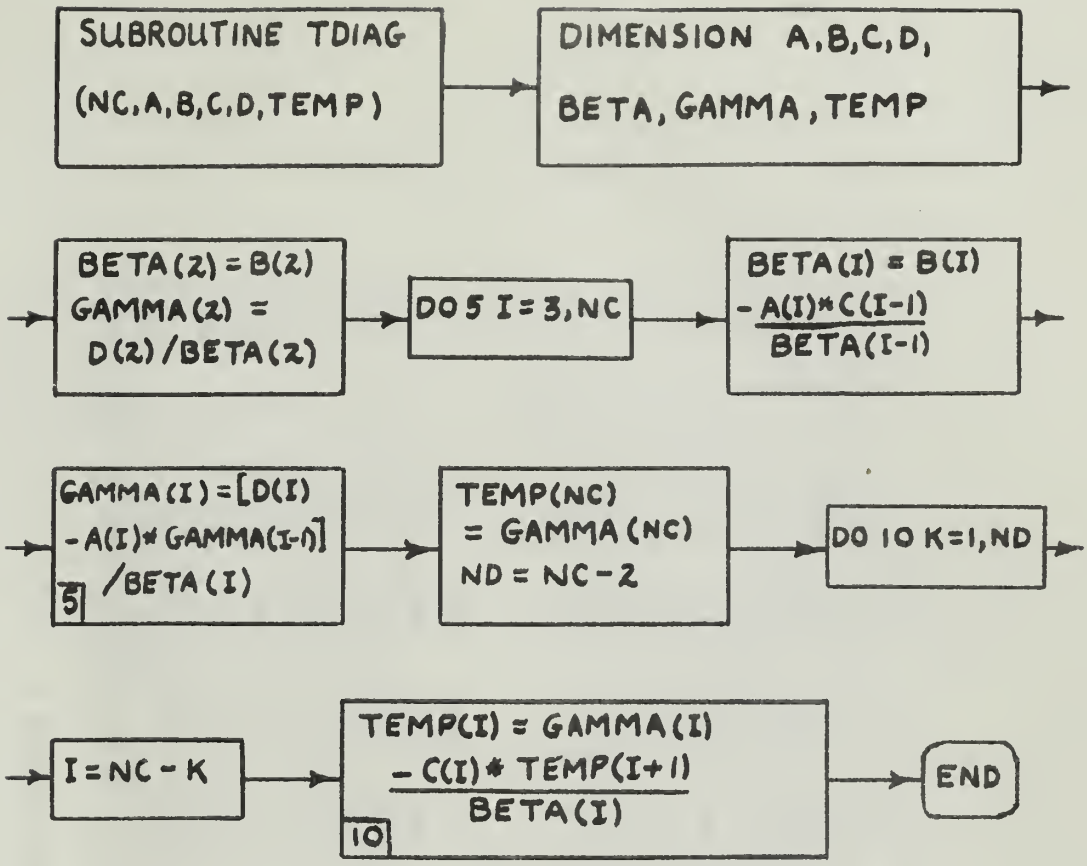

FIGURE 2 FLOWCHART FOR ALGORITHM SUBROUTINE

TIME = 0.250

NUMERICAL SOLUTION

Y \ X	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
0	.4693	.4636	.4466	.4187	.3805	.3329	.2770	.2141	.1458	.0738	0
.1	.4636	.4580	.4412	.4136	.3759	.3288	.2736	.2115	.1441	.0729	0
.2	.4466	.4412	.4250	.3984	.3621	.3168	.2636	.2037	.1388	.0703	0
.3	.4187	.4136	.3984	.3735	.3394	.2970	.2471	.1910	.1301	.0659	0
.4	.3805	.3759	.3621	.3394	.3084	.2699	.2245	.1736	.1182	.0599	0
.5	.3329	.3288	.3168	.2970	.2699	.2361	.1964	.1519	.1034	.0524	0
.6	.2770	.2736	.2636	.2471	.2245	.1964	.1634	.1264	.0861	.0436	0
.7	.2141	.2115	.2037	.1910	.1736	.1519	.1264	.0977	.0665	.0337	0
.8	.1458	.1441	.1388	.1301	.1182	.1034	.0861	.0665	.0453	.0229	0
.9	.0738	.0729	.0703	.0659	.0599	.0524	.0436	.0337	.0229	.0116	0
1.0	0	0	0	0	0	0	0	0	0	0	0

ANALYTIC SOLUTION

Y \ X	0	.1	.2	.3	.4	.5	.6	.7	.8	.9
0	.4698	.4642	.4472	.4195	.3814	.3338	.2779	.2149	.1464	.0742
.1	.4642	.4586	.4418	.4144	.3768	.3298	.2745	.2123	.1447	.0733
.2	.4472	.4418	.4257	.3993	.3630	.3178	.2645	.2046	.1394	.0706
.3	.4195	.4144	.3993	.3745	.3405	.2980	.2481	.1919	.1307	.0662
.4	.3814	.3768	.3630	.3405	.3096	.2710	.2256	.1745	.1189	.0602
.5	.3338	.3298	.3178	.2980	.2710	.2372	.1974	.1527	.1041	.0527
.6	.2779	.2745	.2645	.2481	.2256	.1974	.1664	.1271	.0866	.0439
.7	.2149	.2123	.2046	.1919	.1745	.1527	.1271	.0983	.0670	.0339
.8	.1464	.1447	.1394	.1307	.1189	.1041	.0866	.0670	.0456	.0231
.9	.0742	.0733	.0706	.0662	.0602	.0527	.0439	.0339	.0231	.0117

FIGURE 3 SOLUTION PRINTOUT FOR PROGRAM IADMT, EXAMPLE 1

TIME = 0.250

DIFFERENCE

Y										
0	.0005	.0005	.0006	.0008	.0009	.0009	.0009	.0008	.0006	.0004
.1	.0005	.0006	.0007	.0008	.0009	.0010	.0010	.0008	.0006	.0004
.2	.0006	.0007	.0007	.0009	.0010	.0010	.0010	.0009	.0006	.0004
.3	.0008	.0008	.0009	.0010	.0011	.0011	.0010	.0009	.0006	.0004
.4	.0009	.0009	.0010	.0011	.0011	.0011	.0011	.0009	.0006	.0004
.5	.0009	.0010	.0010	.0011	.0011	.0011	.0010	.0009	.0006	.0003
.6	.0009	.0010	.0010	.0010	.0011	.0010	.0009	.0008	.0006	.0003
.7	.0008	.0008	.0009	.0009	.0009	.0009	.0008	.0006	.0005	.0003
.8	.0006	.0006	.0006	.0006	.0006	.0006	.0006	.0005	.0003	.0002
.9	.0004	.0004	.0004	.0004	.0004	.0003	.0003	.0003	.0002	.0001

PERCENT DIFFERENCE

0	.11	.11	.14	.18	.23	.28	.34	.38	.41	.47
.1	.11	.12	.15	.19	.24	.29	.35	.39	.42	.49
.2	.14	.15	.18	.22	.27	.32	.37	.42	.45	.52
.3	.18	.19	.22	.26	.31	.36	.41	.46	.49	.56
.4	.23	.24	.27	.31	.36	.41	.47	.51	.54	.60
.5	.28	.29	.32	.36	.41	.46	.51	.56	.59	.65
.6	.34	.35	.37	.41	.47	.51	.57	.61	.65	.70
.7	.38	.39	.42	.46	.51	.56	.61	.65	.69	.75
.8	.41	.42	.45	.49	.54	.59	.65	.69	.72	.80
.9	.47	.49	.52	.56	.60	.65	.70	.75	.80	.88

X →	0	.1	.2	.3	.4	.5	.6	.7	.8	.9

FIGURE 4 SOLUTION PRINTOUT FOR PROGRAM IADMT, EXAMPLE 1

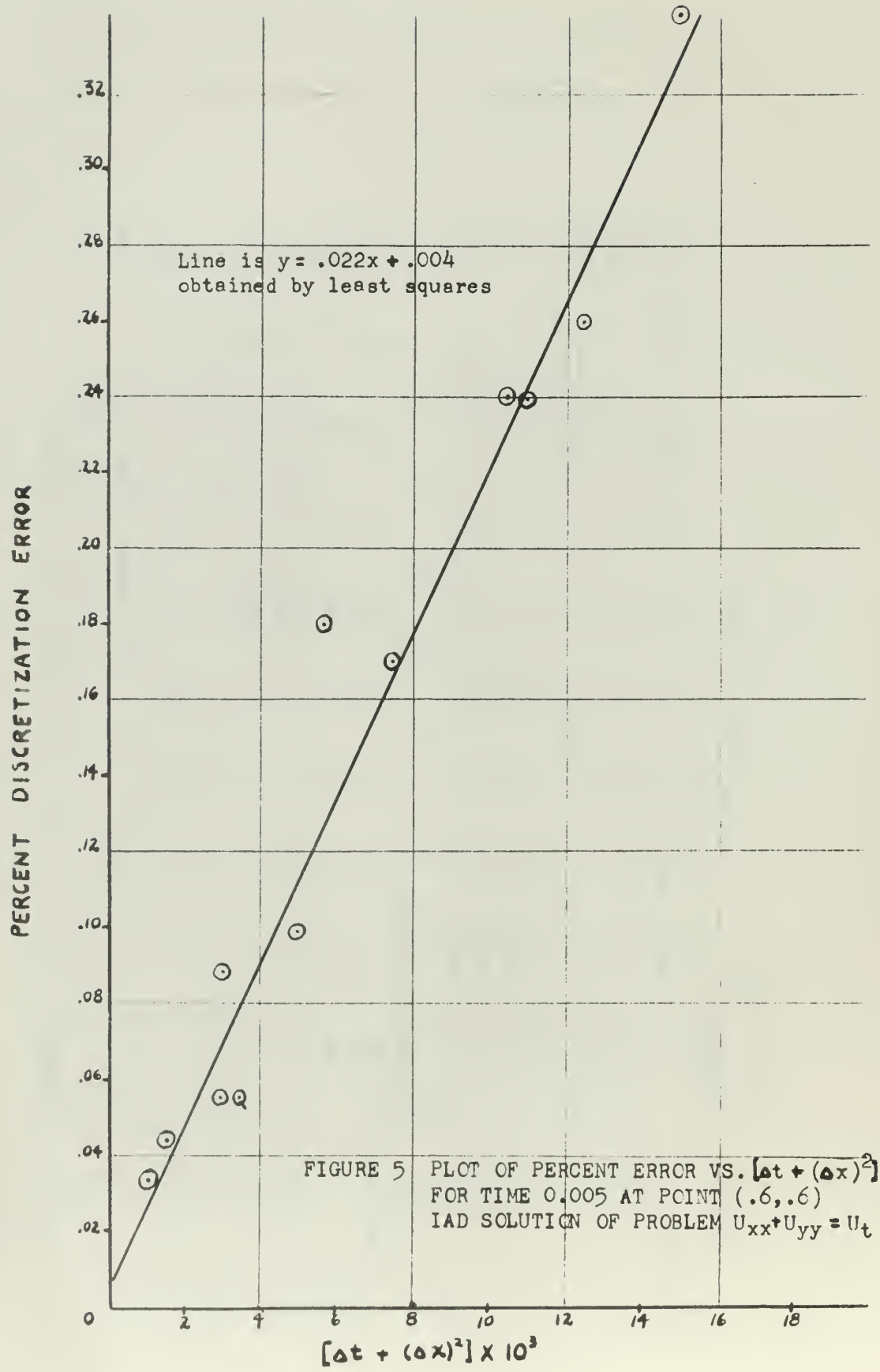

FIGURE 5 PLOT OF PERCENT ERROR VS. $[\Delta t + (\Delta x)^2]$
FOR TIME 0.005 AT POINT (.6,.6)
IAD SOLUTION OF PROBLEM $U_{xx}+U_{yy} = U_t$

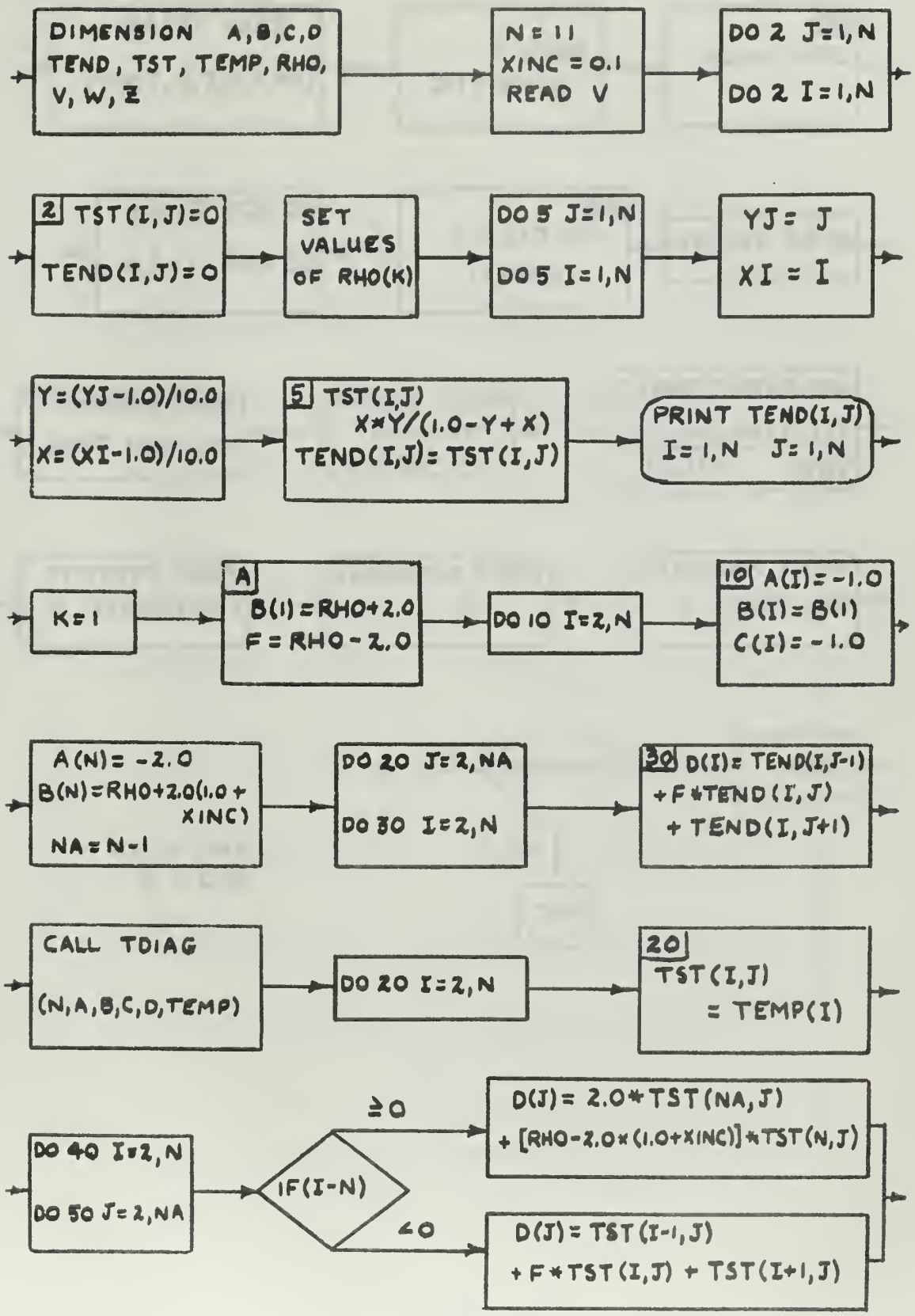

FIGURE 6 FLOWCHART FOR PROGRAM IADM

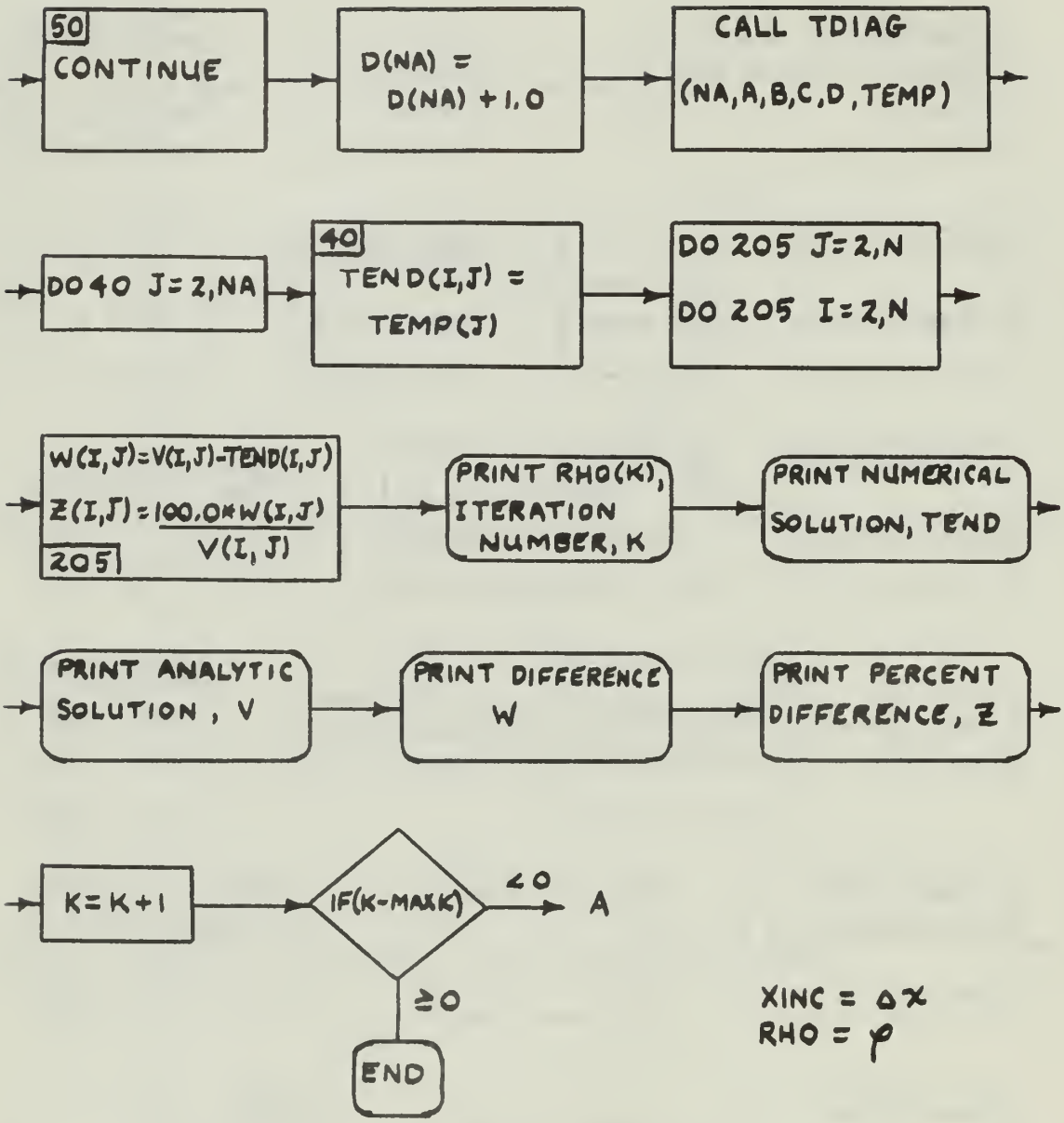

FIGURE 6 (CONTINUED)

RHO = 3.7037 ITERATION NUMBER 1

NUMERICAL SOLUTION

Y	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0101	.0191	.0265	.0327	.0379	.0423	.0462	.0494	.0515	.0494
.2	0	.0218	.0408	.0563	.0691	.0798	.0888	.0965	.1029	.1071	.1026
.3	0	.0361	.0670	.0919	.1119	.1284	.1422	.1538	.1634	.1694	.1622
.4	0	.0542	.0999	.1355	.1635	.1860	.2046	.2200	.2326	.2402	.2296
.5	0	.0784	.1425	.1904	.2269	.2556	.2786	.2973	.3126	.3215	.3067
.6	0	.1124	.2001	.2621	.3071	.3413	.3681	.3896	.4064	.4156	.3957
.7	0	.1651	.2833	.3597	.4118	.4496	.4783	.5007	.5178	.5260	.4998
.8	0	.2603	.4145	.5005	.5539	.5903	.6169	.6371	.6519	.6572	.6241
.9	0	.4708	.6409	.7130	.7514	.7757	.7928	.8054	.8140	.8145	.7791
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

ANALYTIC SOLUTION

	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0143	.0278	.0396	.0493	.0566	.0613	.0635	.0633	.0608	.0562
.2	0	.0296	.0573	.0814	.1011	.1157	.1251	.1294	.1288	.1237	.1143
.3	0	.0468	.0903	.1279	.1580	.1801	.1940	.2002	.1990	.1909	.1763
.4	0	.0674	.1294	.1820	.2231	.2526	.2707	.2783	.2761	.2646	.2443
.5	0	.0938	.1783	.2477	.3002	.3364	.3579	.3663	.3625	.3470	.3203
.6	0	.1300	.2431	.3308	.3935	.4348	.4582	.4664	.4605	.4407	.4069
.7	0	.1851	.3345	.4396	.5084	.5507	.5734	.5803	.5724	.5485	.5068
.8	0	.2821	.4731	.5845	.6494	.6859	.7040	.7086	.7002	.6745	.6248
.9	0	.4945	.6911	.7730	.8208	.8391	.8476	.8486	.8442	.8255	.7692
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

X → 0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0

FIGURE 7 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO = 3.7037 ITERATION NUMBER 1

Y

DIFFERENCE

.1	.0042	.0087	.0131	.0166	.0187	.0189	.0173	.0139	.0093	.0068
.2	.0078	.0165	.0251	.0320	.0359	.0363	.0329	.0259	.0166	.0117
.3	.0107	.0233	.0360	.0461	.0517	.0518	.0464	.0356	.0215	.0141
.4	.0132	.0295	.0465	.0596	.0665	.0661	.0583	.0435	.0243	.0147
.5	.0154	.0358	.0573	.0732	.0808	.0793	.0688	.0498	.0255	.0136
.6	.0176	.0429	.0688	.0864	.0935	.0901	.0768	.0541	.0251	.0112
.7	.0200	.0512	.0799	.0966	.1012	.0952	.0796	.0546	.0225	.0070
.8	.0218	.0586	.0839	.0956	.0956	.0872	.0716	.0484	.0174	.0007
.9	.0237	.0502	.0601	.0694	.0634	.0548	.0433	.0302	.0110	-.0100
1.0	0	0	0	0	0	0	0	0	0	0

PERCENT DIFFERENCE

.1	29.24	31.37	33.14	33.75	33.02	30.92	27.28	21.96	15.30	12.08
.2	26.40	28.83	30.87	31.67	31.06	29.01	25.41	20.11	13.42	10.19
.3	22.95	25.76	28.16	29.17	28.69	26.72	23.17	17.90	11.24	8.01
.4	19.54	22.79	25.54	26.73	26.34	24.43	20.95	15.76	9.19	6.01
.5	16.40	20.09	23.12	24.39	24.02	22.15	18.77	13.75	7.36	4.26
.6	13.57	17.67	20.79	21.96	21.50	19.66	16.47	11.74	5.70	2.74
.7	10.82	15.32	18.17	19.00	18.37	16.60	13.72	9.54	4.11	1.39
.8	7.73	12.38	14.36	14.72	13.94	12.38	10.10	6.91	2.58	0.12
.9	4.80	7.26	7.77	8.46	7.55	6.46	5.10	3.58	1.33	-1.30
1.0	0	0	0	0	0	0	0	0	0	0

X →	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0

FIGURE 8 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO = 2.1978 ITERATION NUMBER 2

NUMERICAL SOLUTION

Y \ X	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0104	.0199	.0280	.0348	.0404	.0450	.0487	.0512	.0514	.0483
.2	0	.0218	.0418	.0587	.0726	.0840	.0934	.1008	.1057	.1059	.0995
.3	0	.0356	.0678	.0948	.1165	.1340	.1482	.1593	.1665	.1664	.1562
.4	0	.0531	.1004	.1391	.1694	.1933	.2122	.2268	.2360	.2352	.2205
.5	0	.0767	.1433	.1956	.2352	.2654	.2887	.3063	.3168	.3147	.2947
.6	0	.1109	.2028	.2710	.3198	.3553	.3819	.4013	.4122	.4078	.3815
.7	0	.1663	.2917	.3760	.4316	.4698	.4971	.5164	.5261	.5183	.4846
.8	0	.2681	.4330	.5269	.5817	.6164	.6400	.6559	.6624	.6511	.6102
.9	0	.4807	.6601	.7370	.7750	.7969	.8109	.8198	.8224	.8112	.7704
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

ANALYTIC SOLUTION

Y \ X	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0143	.0278	.0396	.0493	.0566	.0613	.0635	.0633	.0608	.0562
.2	0	.0296	.0573	.0814	.1011	.1157	.1251	.1294	.1288	.1237	.1143
.3	0	.0468	.0903	.1279	.1580	.1801	.1940	.2002	.1990	.1909	.1763
.4	0	.0674	.1294	.1820	.2231	.2526	.2707	.2783	.2761	.2646	.2443
.5	0	.0938	.1783	.2477	.3002	.3364	.3579	.3663	.3625	.3470	.3203
.6	0	.1300	.2431	.3308	.3935	.4348	.4582	.4664	.4605	.4407	.4069
.7	0	.1851	.3345	.4396	.5084	.5507	.5734	.5803	.5724	.5485	.5068
.8	0	.2821	.4731	.5845	.6494	.6859	.7040	.7086	.7002	.6745	.6248
.9	0	.4945	.6911	.7730	.8208	.8391	.8476	.8486	.8442	.8255	.7692
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

FIGURE 9 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO = 2.1978 ITERATION NUMBER 2

DIFFERENCE

Y	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
.1	.0040	.0079	.0116	.0145	.0162	.0163	.0147	.0121	.0094	.0079
.2	.0077	.0155	.0227	.0285	.0317	.0317	.0286	.0231	.0178	.0148
.3	.0112	.0225	.0331	.0415	.0461	.0459	.0409	.0325	.0245	.0202
.4	.0143	.0290	.0429	.0537	.0593	.0585	.0515	.0401	.0294	.0238
.5	.0171	.0350	.0521	.0649	.0710	.0692	.0600	.0456	.0323	.0256
.6	.0191	.0402	.0598	.0738	.0795	.0763	.0651	.0483	.0329	.0254
.7	.0188	.0428	.0635	.0768	.0810	.0763	.0640	.0464	.0302	.0223
.8	.0140	.0400	.0576	.0677	.0694	.0640	.0528	.0378	.0234	.0146
.9	.0138	.0310	.0360	.0458	.0422	.0366	.0288	.0218	.0142	-.0012
1.0	0	0	0	0	0	0	0	0	0	0

PERCENT DIFFERENCE

.1	27.77	28.54	29.29	29.45	28.59	26.53	23.22	19.05	15.47	14.05
.2	26.18	27.04	27.90	28.15	27.36	25.34	22.08	17.94	14.38	12.96
.3	23.90	24.89	25.90	26.28	25.58	23.64	20.43	16.35	12.84	11.44
.4	21.23	22.40	23.58	24.09	23.48	21.62	18.51	14.53	11.11	9.74
.5	18.25	19.66	21.02	21.63	21.10	19.34	16.38	12.59	9.31	8.00
.6	14.71	16.55	18.09	18.75	18.28	16.66	13.95	10.48	7.47	6.24
.7	10.17	12.80	14.45	15.10	14.70	13.31	11.02	8.10	5.51	4.39
.8	4.96	8.46	9.85	10.43	10.12	9.09	7.45	5.40	3.47	2.34
.9	2.80	4.48	4.66	5.58	5.03	4.32	3.39	2.58	1.72	-.16
1.0	0	0	0	0	0	0	0	0	0	0

X → .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0

FIGURE 10 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO = 0.7633 ITERATION NUMBER 3

NUMERICAL SOLUTION

Y											
0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0112	.0217	.0310	.0389	.0452	.0498	.0525	.0532	.0517	.0479
.2	0	.0234	.0452	.0646	.0807	.0933	.1025	.1080	.1092	.1060	.0983
.3	0	.0378	.0729	.1034	.1284	.1477	.1614	.1695	.1710	.1657	.1535
.4	0	.0563	.1077	.1512	.1859	.2119	.2300	.2402	.2415	.2337	.2164
.5	0	.0816	.1540	.2128	.2578	.2902	.3120	.3236	.3241	.3130	.2898
.6	0	.1186	.2187	.2951	.3499	.3875	.4115	.4233	.4222	.4071	.3768
.7	0	.1771	.3131	.4067	.4683	.5077	.5314	.5419	.5386	.5194	.4814
.8	0	.2796	.4552	.5577	.6175	.6526	.6723	.6799	.6749	.6536	.6088
.9	0	.4883	.6745	.7568	.7976	.8194	.8309	.8347	.8303	.8132	.7701
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

ANALYTIC SOLUTION

0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0143	.0278	.0396	.0493	.0566	.0613	.0635	.0633	.0608	.0562
.2	0	.0296	.0573	.0814	.1011	.1157	.1251	.1294	.1288	.1237	.1143
.3	0	.0468	.0903	.1279	.1580	.1801	.1940	.2002	.1990	.1909	.1763
.4	0	.0674	.1294	.1820	.2231	.2526	.2707	.2783	.2761	.2646	.2443
.5	0	.0938	.1783	.2477	.3002	.3364	.3579	.3663	.3625	.3470	.3203
.6	0	.1300	.2431	.3308	.3935	.4348	.4582	.4664	.4605	.4407	.4069
.7	0	.1851	.3345	.4396	.5084	.5507	.5734	.5803	.5724	.5485	.5068
.8	0	.2821	.4731	.5845	.6494	.6859	.7040	.7086	.7002	.6745	.6248
.9	0	.4945	.6911	.7730	.8208	.8391	.8476	.8486	.8442	.8255	.7692
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

X										
0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0

FIGURE 11 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO = 0.7633 ITERATION NUMBER 3

DIFFERENCE

Y	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
.1	.0032	.0061	.0086	.0104	.0114	.0115	.0109	.0101	.0091	.0083
.2	.0062	.0120	.0169	.0205	.0224	.0226	.0214	.0196	.0177	.0160
.3	.0090	.0174	.0245	.0296	.0324	.0326	.0307	.0280	.0252	.0228
.4	.0111	.0217	.0307	.0373	.0407	.0407	.0381	.0346	.0309	.0279
.5	.0122	.0243	.0349	.0424	.0461	.0459	.0427	.0383	.0340	.0306
.6	.0115	.0244	.0358	.0437	.0473	.0467	.0431	.0383	.0336	.0300
.7	.0080	.0215	.0329	.0401	.0430	.0420	.0384	.0339	.0291	.0254
.8	.0025	.0178	.0267	.0319	.0332	.0318	.0287	.0254	.0210	.0160
.9	.0062	.0165	.0163	.0232	.0196	.0167	.0139	.0139	.0122	-.0009
1.0	0	0	0	0	0	0	0	0	0	0

PERCENT DIFFERENCE

.1	22.15	22.02	21.69	21.11	20.17	18.81	17.23	15.90	14.97	14.69
.2	21.04	20.98	20.73	20.23	19.35	18.04	16.52	15.21	14.30	14.02
.3	19.17	19.23	19.13	18.76	17.99	16.79	15.34	14.08	13.20	12.92
.4	16.51	16.78	16.90	16.71	16.11	15.03	13.70	12.52	11.68	11.42
.5	13.03	13.65	14.07	14.13	13.71	12.82	11.64	10.58	9.80	9.55
.6	8.83	10.03	10.82	11.10	10.88	10.20	9.23	8.32	7.62	7.38
.7	4.34	6.42	7.48	7.88	7.80	7.33	6.62	5.92	5.30	5.01
.8	0.87	3.77	4.57	4.91	4.84	4.51	4.05	3.63	3.11	2.57
.9	1.26	2.39	2.11	2.83	2.34	1.97	1.64	1.64	1.48	-0.12
1.0	0	0	0	0	0	0	0	0	0	0

X	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0

FIGURE 12 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO = 0.2180 ITERATION NUMBER 4

NUMERICAL SOLUTION

Y											
0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0136	.0264	.0375	.0466	.0533	.0576	.0594	.0591	.0567	.0524
.2	0	.0283	.0547	.0775	.0960	.1095	.1181	.1218	.1210	.1161	.1072
.3	0	.0452	.0869	.1227	.1511	.1717	.1845	.1899	.1885	.1806	.1667
.4	0	.0659	.1259	.1762	.2152	.2429	.2597	.2665	.2641	.2529	.2335
.5	0	.0928	.1752	.2420	.2920	.3264	.3466	.3543	.3504	.3354	.3096
.6	0	.1303	.2410	.3256	.3857	.4252	.4476	.4553	.4496	.4305	.3976
.7	0	.1878	.3333	.4345	.5009	.5419	.5641	.5709	.5634	.5406	.5002
.8	0	.2877	.4705	.5786	.6420	.6783	.6969	.7017	.6935	.6695	.6228
.9	0	.4926	.6827	.7679	.8107	.8332	.8440	.8464	.8403	.8217	.7776
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

ANALYTIC SOLUTION

0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0143	.0278	.0396	.0493	.0566	.0613	.0635	.0633	.0608	.0562
.2	0	.0296	.0573	.0814	.1011	.1157	.1251	.1294	.1288	.1237	.1143
.3	0	.0468	.0903	.1279	.1580	.1801	.1940	.2002	.1990	.1909	.1763
.4	0	.0674	.1294	.1820	.2231	.2526	.2707	.2783	.2761	.2646	.2443
.5	0	.0938	.1783	.2477	.3002	.3364	.3579	.3663	.3625	.3470	.3203
.6	0	.1300	.2431	.3308	.3935	.4348	.4582	.4664	.4605	.4407	.4069
.7	0	.1851	.3345	.4396	.5084	.5507	.5734	.5803	.5724	.5485	.5068
.8	0	.2821	.4731	.5845	.6494	.6859	.7040	.7086	.7002	.6745	.6248
.9	0	.4945	.6911	.7730	.8208	.8391	.8476	.8486	.8442	.8255	.7692
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

X	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0

FIGURE 13 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO = 0.2180 ITERATION NUMBER 4

DIFFERENCE

Y	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
.1	.0007	.0014	.0021	.0027	.0033	.0037	.0040	.0041	.0041	.0038
.2	.0013	.0026	.0039	.0052	.0062	.0070	.0076	.0078	.0076	.0071
.3	.0016	.0034	.0052	.0069	.0084	.0095	.0103	.0105	.0103	.0096
.4	.0016	.0036	.0058	.0079	.0097	.0110	.0118	.0120	.0116	.0108
.5	.0010	.0031	.0057	.0081	.0100	.0113	.0120	.0121	.0116	.0107
.6	-.0003	.0021	.0052	.0078	.0096	.0106	.0111	.0109	.0102	.0093
.7	-.0027	.0012	.0051	.0076	.0088	.0093	.0095	.0090	.0079	.0066
.8	-.0056	.0026	.0058	.0075	.0075	.0072	.0070	.0068	.0051	.0020
.9	.0019	.0084	.0051	.0101	.0059	.0036	.0023	.0039	.0037	-.0084
1.0	0	0	0	0	0	0	0	0	0	0

PERCENT DIFFERENCE

.1	4.90	5.08	5.32	5.57	5.81	6.09	6.37	6.55	6.68	6.75
.2	4.32	4.55	4.83	5.10	5.35	5.62	5.90	6.06	6.17	6.24
.3	3.44	3.73	4.08	4.39	4.66	4.92	5.16	5.29	5.38	5.44
.4	2.33	2.74	3.18	3.55	3.82	4.06	4.25	4.34	4.40	4.44
.5	1.07	1.72	2.29	2.70	2.97	3.15	3.28	3.33	3.33	3.35
.6	-0.22	0.85	1.58	1.99	2.20	2.32	2.38	2.37	2.31	2.29
.7	-1.43	0.37	1.17	1.49	1.59	1.63	1.63	1.58	1.44	1.31
.8	-1.98	0.55	1.00	1.15	1.10	1.02	0.98	0.97	0.75	0.31
.9	0.39	1.21	0.66	1.23	0.70	0.42	0.27	0.46	0.45	-1.09
1.0	0	0	0	0	0	0	0	0	0	0

X → .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0

FIGURE 14 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO= 0.0246 ITERATION NUMBER 5

NUMERICAL SOLUTION

Y	X = 0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0143	.0277	.0395	.0491	.0562	.0608	.0629	.0626	.0601	.0555
.2	0	.0296	.0572	.0812	.1006	.1150	.1241	.1282	.1276	.1224	.1131
.3	0	.0469	.0904	.1277	.1574	.1790	.1926	.1985	.1973	.1892	.1747
.4	0	.0679	.1298	.1818	.2224	.2513	.2690	.2764	.2741	.2627	.2425
.5	0	.0948	.1792	.2477	.2993	.3348	.3560	.3642	.3605	.3453	.3187
.6	0	.1322	.2446	.3308	.3923	.4328	.4561	.4643	.4588	.4394	.4058
.7	0	.1893	.3363	.4387	.5063	.5483	.5711	.5783	.5710	.5480	.5070
.8	0	.2888	.4726	.5817	.6459	.6829	.7019	.7071	.6990	.6748	.6278
.9	0	.4931	.6838	.7696	.8127	.8356	.8467	.8492	.8431	.8245	.7802
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

ANALYTIC SOLUTION

Y	X = 0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0143	.0278	.0396	.0493	.0566	.0613	.0635	.0633	.0608	.0562
.2	0	.0296	.0573	.0814	.1011	.1157	.1251	.1294	.1288	.1237	.1143
.3	0	.0468	.0903	.1279	.1580	.1801	.1940	.2002	.1990	.1909	.1763
.4	0	.0674	.1294	.1820	.2231	.2526	.2707	.2783	.2761	.2646	.2443
.5	0	.0938	.1783	.2477	.3002	.3364	.3579	.3663	.3625	.3470	.3203
.6	0	.1300	.2431	.3308	.3935	.4348	.4582	.4664	.4605	.4407	.4069
.7	0	.1851	.3345	.4396	.5084	.5507	.5734	.5803	.5724	.5485	.5068
.8	0	.2821	.4731	.5845	.6494	.6859	.7040	.7086	.7002	.6745	.6248
.9	0	.4945	.6911	.7730	.8208	.8391	.8476	.8486	.8442	.8255	.7692
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

FIGURE 15 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO = 0.0246 ITERATION NUMBER 5

DIFFERENCE

Y	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
.1	.0000	.0001	.0001	.0003	.0004	.0005	.0006	.0007	.0007	.0007
.2	.0000	.0001	.0002	.0005	.0007	.0010	.0012	.0013	.0013	.0012
.3	-.0002	-.0001	.0002	.0006	.0010	.0014	.0016	.0017	.0017	.0016
.4	-.0004	-.0004	.0001	.0008	.0013	.0017	.0019	.0020	.0019	.0018
.5	-.0010	-.0009	.0000	.0009	.0015	.0019	.0021	.0020	.0017	.0016
.6	-.0021	-.0015	.0000	.0013	.0019	.0021	.0021	.0017	.0013	.0010
.7	-.0042	-.0017	.0008	.0021	.0024	.0023	.0020	.0015	.0005	-.0002
.8	-.0067	.0004	.0028	.0035	.0030	.0021	.0016	.0013	-.0003	-.0030
.9	.0013	.0072	.0035	.0081	.0035	.0009	-.0006	.0010	.0010	-.0110
1.0	0	0	0	0	0	0	0	0	0	0

PERCENT DIFFERENCE

.1	0.06	0.20	0.37	0.55	0.69	0.85	0.99	1.07	1.13	1.18
.2	0.09	0.09	0.30	0.49	0.65	0.79	0.92	0.99	1.04	1.09
.3	-0.32	-0.08	0.19	0.41	0.58	0.71	0.82	0.86	0.90	0.93
.4	-0.65	-0.29	0.07	0.34	0.51	0.62	0.69	0.71	0.71	0.74
.5	-1.09	-0.50	-0.01	0.30	0.46	0.54	0.56	0.54	0.50	0.50
.6	-1.64	-0.63	0.01	0.32	0.44	0.46	0.45	0.38	0.29	0.25
.7	-2.26	-0.52	0.19	0.42	0.44	0.40	0.34	0.25	0.09	-0.04
.8	-2.37	0.09	0.48	0.55	0.43	0.30	0.22	0.18	-0.04	-0.48
.9	0.27	1.05	0.45	0.98	0.42	0.11	-0.07	0.12	0.12	-1.43
1.0	0	0	0	0	0	0	0	0	0	0

X → .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0

FIGURE 16 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHC = 0.0246 ITERATION NUMBER 10

NUMERICAL SOLUTION

Y											
0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0144	.0279	.0397	.0493	.0565	.0611	.0632	.0630	.0605	.0559
.2	0	.0298	.0575	.0816	.1011	.1155	.1247	.1289	.1283	.1232	.1138
.3	0	.0471	.0907	.1282	.1580	.1797	.1934	.1995	.1983	.1902	.1756
.4	0	.0681	.1302	.1824	.2231	.2520	.2699	.2774	.2752	.2638	.2435
.5	0	.0950	.1796	.2483	.2999	.3355	.3568	.3652	.3615	.3463	.3197
.6	0	.1323	.2449	.3313	.3928	.4334	.4567	.4651	.4596	.4403	.4066
.7	0	.1894	.3366	.4391	.5067	.5487	.5717	.5790	.5717	.5487	.5077
.8	0	.2889	.4728	.5819	.6462	.6832	.7023	.7075	.6994	.6753	.6282
.9	0	.4932	.6839	.7697	.8129	.8357	.8469	.8494	.8434	.8248	.7804
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

ANALYTIC SOLUTION

0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0143	.0278	.0396	.0493	.0566	.0613	.0635	.0633	.0608	.0562
.2	0	.0296	.0573	.0814	.1011	.1157	.1251	.1294	.1288	.1237	.1143
.3	0	.0468	.0903	.1279	.1580	.1801	.1940	.2002	.1990	.1909	.1763
.4	0	.0674	.1294	.1820	.2231	.2526	.2707	.2783	.2761	.2646	.2443
.5	0	.0938	.1783	.2477	.3002	.3364	.3579	.3663	.3625	.3470	.3203
.6	0	.1300	.2431	.3308	.3935	.4348	.4582	.4664	.4605	.4407	.4069
.7	0	.1851	.3345	.4396	.5084	.5507	.5734	.5803	.5724	.5485	.5068
.8	0	.2821	.4731	.5845	.6494	.6859	.7040	.7086	.7002	.6745	.6248
.9	0	.4945	.6911	.7730	.8208	.8391	.8476	.8486	.8442	.8255	.7692
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

X →	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0

FIGURE 17 SOLUTION PRINTOUT FOR PROGRAM IADE, EXAMPLE 2

RHO = 0.0246 ITERATION NUMBER 10

Y

DIFFERENCE

.1	-.0001	-.0001	-.0001	.0000	.0001	.0002	.0002	.0003	.0003	.0003
.2	-.0002	-.0002	-.0001	.0000	.0002	.0004	.0005	.0005	.0005	.0005
.3	-.0003	-.0004	-.0003	.0000	.0004	.0006	.0007	.0007	.0007	.0007
.4	-.0007	-.0008	-.0004	.0001	.0005	.0008	.0009	.0009	.0008	.0008
.5	-.0012	-.0013	-.0006	.0003	.0009	.0011	.0011	.0009	.0007	.0007
.6	-.0023	-.0019	-.0004	.0008	.0014	.0015	.0013	.0009	.0004	.0002
.7	-.0043	-.0020	.0005	.0017	.0020	.0018	.0014	.0008	-.0002	-.0008
.8	-.0068	.0003	.0025	.0033	.0026	.0018	.0011	.0008	-.0007	-.0034
.9	.0013	.0071	.0034	.0079	.0033	.0007	-.0008	.0008	.0007	-.0112
1.0	0	0	0	0	0	0	0	0	0	0

PERCENT DIFFERENCE

.1	-0.49	-0.34	-0.16	0.04	0.20	0.32	0.38	0.42	0.46	0.51
.2	-0.60	-0.40	-0.18	0.03	0.20	0.30	0.36	0.40	0.43	0.47
.3	-0.74	-0.49	-0.22	0.02	0.20	0.30	0.34	0.36	0.37	0.41
.4	-0.97	-0.60	-0.24	0.04	0.22	0.30	0.32	0.31	0.30	0.33
.5	-1.30	-0.72	-0.22	0.09	0.26	0.31	0.30	0.26	0.21	0.21
.6	-1.77	-0.77	-0.12	0.19	0.31	0.32	0.28	0.19	0.09	0.06
.7	-2.33	-0.60	0.11	0.34	0.36	0.31	0.24	0.14	-0.03	-0.16
.8	-2.40	0.05	0.43	0.50	0.38	0.25	0.16	0.12	-0.11	-0.55
.9	0.26	1.03	0.43	0.97	0.40	0.08	-0.10	0.09	0.08	-1.46
1.0	0	0	0	0	0	0	0	0	0	0

X → .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0

FIGURE 18 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO = 0.2500 ITERATION NUMBER 5

NUMERICAL SOLUTION

Y											
0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0139	.0269	.0382	.0474	.0543	.0587	.0607	.0605	.0580	.0536
.2	0	.0288	.0556	.0788	.0975	.1113	.1202	.1242	.1235	.1185	.1095
.3	0	.0458	.0881	.1244	.1532	.1741	.1872	.1930	.1917	.1839	.1698
.4	0	.0665	.1271	.1780	.2175	.2455	.2627	.2699	.2676	.2564	.2367
.5	0	.0934	.1764	.2437	.2940	.3287	.3493	.3573	.3536	.3386	.3126
.6	0	.1308	.2419	.3269	.3872	.4269	.4495	.4576	.4521	.4330	.3999
.7	0	.1881	.3340	.4354	.5020	.5432	.5656	.5726	.5652	.5425	.5019
.8	0	.2879	.4710	.5793	.6428	.6792	.6979	.7029	.6948	.6708	.6241
.9	0	.4927	.6829	.7683	.8111	.8336	.8445	.8470	.8410	.8224	.7782
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

ANALYTIC SOLUTION

0	0	0	0	0	0	0	0	0	0	0	0
.1	0	.0143	.0278	.0396	.0493	.0566	.0613	.0635	.0633	.0608	.0562
.2	0	.0296	.0573	.0814	.1011	.1157	.1251	.1294	.1288	.1237	.1143
.3	0	.0468	.0903	.1279	.1580	.1801	.1940	.2002	.1990	.1909	.1763
.4	0	.0674	.1294	.1820	.2231	.2526	.2707	.2783	.2761	.2646	.2443
.5	0	.0938	.1783	.2477	.3002	.3364	.3579	.3663	.3625	.3470	.3203
.6	0	.1300	.2431	.3308	.3935	.4348	.4582	.4664	.4605	.4407	.4069
.7	0	.1851	.3345	.4396	.5084	.5507	.5734	.5803	.5724	.5485	.5068
.8	0	.2821	.4731	.5845	.6494	.6859	.7040	.7086	.7002	.6745	.6248
.9	0	.4945	.6911	.7730	.8208	.8391	.8476	.8486	.8442	.8255	.7692
1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
X →	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0

FIGURE 19 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

RHO = 0.2500 ITERATION NUMBER 5

DIFFERENCE

Y	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
.1	.0005	.0009	.0014	.0019	.0023	.0026	.0028	.0028	.0027	.0026
.2	.0008	.0017	.0026	.0036	.0044	.0049	.0052	.0053	.0052	.0048
.3	.0010	.0022	.0035	.0048	.0060	.0068	.0072	.0073	.0070	.0066
.4	.0009	.0023	.0040	.0057	.0071	.0080	.0084	.0085	.0082	.0076
.5	.0004	.0019	.0041	.0061	.0077	.0087	.0090	.0089	.0084	.0078
.6	-.0008	.0012	.0040	.0063	.0079	.0086	.0088	.0085	.0077	.0070
.7	-.0030	.0006	.0042	.0064	.0075	.0079	.0077	.0072	.0060	.0049
.8	-.0058	.0021	.0052	.0067	.0066	.0061	.0057	.0054	.0037	.0007
.9	.0018	.0081	.0048	.0097	.0054	.0030	.0016	.0032	.0031	-.0091
1.0	0	0	0	0	0	0	0	0	0	0

PERCENT DIFFERENCE

Y	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
.1	3.14	3.34	3.57	3.84	4.08	4.26	4.36	4.45	4.51	4.56
.2	2.73	2.97	3.24	3.53	3.78	3.95	4.05	4.13	4.19	4.23
.3	2.12	2.41	2.75	3.07	3.33	3.50	3.59	3.65	3.69	3.72
.4	1.34	1.76	2.19	2.55	2.81	2.96	3.03	3.07	3.08	3.11
.5	0.43	1.07	1.64	2.04	2.29	2.42	2.46	2.45	2.42	2.43
.6	-0.58	0.48	1.20	1.61	1.81	1.89	1.88	1.84	1.75	1.72
.7	-1.62	0.17	0.96	1.27	1.37	1.37	1.33	1.26	1.10	0.97
.8	-2.07	0.44	0.89	1.02	0.96	0.87	0.80	0.77	0.55	0.11
.9	0.36	1.18	0.62	1.18	0.65	0.36	0.19	0.38	0.37	-1.18
1.0	0	0	0	0	0	0	0	0	0	0

X → .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0

FIGURE 20 SOLUTION PRINTOUT FOR PROGRAM IADM, EXAMPLE 2

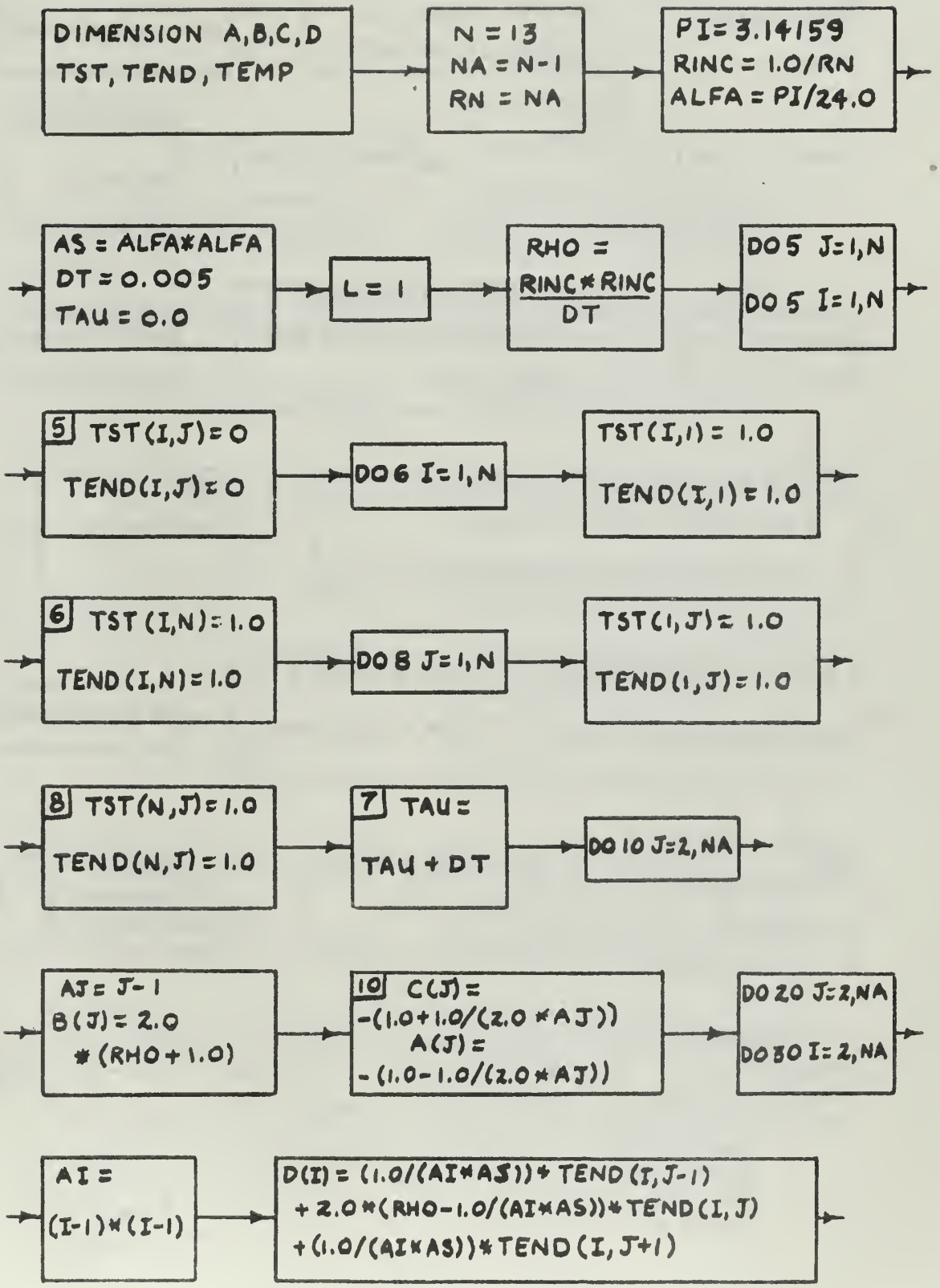

FIGURE 21 FLOWCHART FOR PROGRAM IADMC

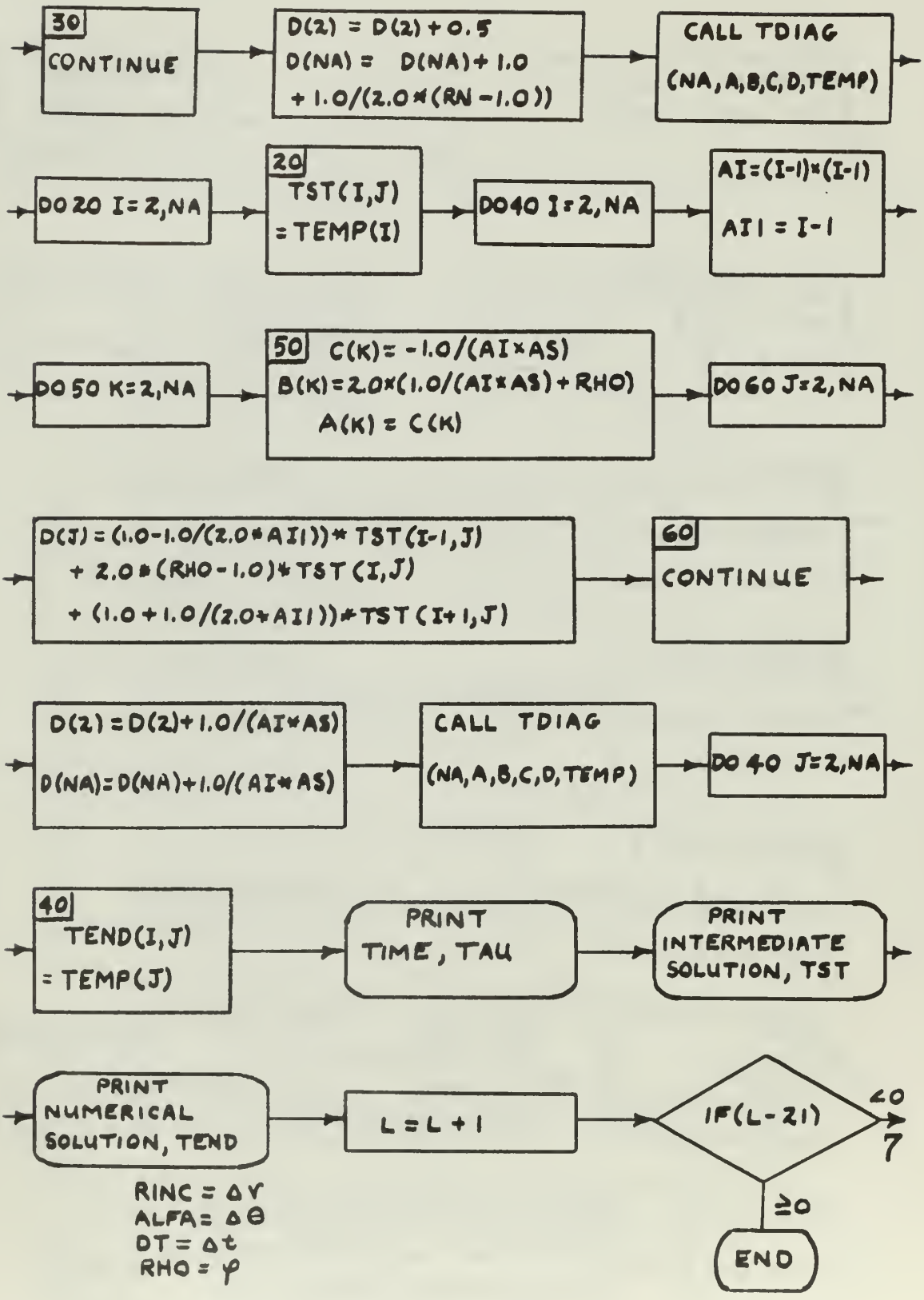

FIGURE 21 (CONTINUED)

TIME 0.100

INTERMEDIATE SOLUTION

θ	r = .083	.166	.250	.333	.416	.500	.583	.666	.750	.833	.916
	.9974	.9932	.9875	.9808	.9745	.9697	.9676	.9688	.9733	.9807	.9901
15°	.9989	.9909	.9785	.9644	.9513	.9417	.9375	.9397	.9485	.9628	.9808
	.9956	.9842	.9675	.9485	.9307	.9174	.9116	.9148	.9272	.9474	.9729
30°	.9957	.9818	.9611	.9375	.9154	.8990	.8917	.8956	.9108	.9356	.9668
	.9951	.9795	.9565	.9302	.9055	.8873	.8793	.8836	.9006	.9282	.9630
45°	.9948	.9787	.9549	.9277	.9022	.8833	.8750	.8795	.8971	.9257	.9617
	.9951	.9795	.9565	.9302	.9055	.8873	.8793	.8836	.9006	.9282	.9630
60°	.9957	.9818	.9611	.9375	.9154	.8990	.8917	.8956	.9108	.9356	.9668
	.9956	.9842	.9675	.9485	.9307	.9174	.9116	.9148	.9272	.9474	.9729
75°	.9989	.9909	.9785	.9644	.9513	.9417	.9375	.9397	.9485	.9628	.9808
	.9974	.9932	.9875	.9808	.9745	.9697	.9676	.9688	.9733	.9807	.9901

NUMERICAL SOLUTION

θ	r = .083	.166	.250	.333	.416	.500	.583	.666	.750	.833	.916
	.9986	.9947	.9889	.9823	.9762	.9717	.9697	.9708	.9750	.9820	.9907
15°	.9974	.9899	.9788	.9661	.9542	.9453	.9415	.9436	.9518	.9652	.9821
	.9962	.9856	.9698	.9519	.9351	.9226	.9172	.9202	.9318	.9508	.9746
30°	.9954	.9824	.9631	.9412	.9206	.9053	.8986	.9023	.9166	.9398	.9690
	.9949	.9804	.9589	.9344	.9114	.8944	.8869	.8911	.9070	.9329	.9654
45°	.9947	.9797	.9574	.9320	.9083	.8907	.8829	.8872	.9037	.9305	.9642
	.9949	.9804	.9589	.9344	.9114	.8944	.8869	.8911	.9070	.9329	.9654
60°	.9954	.9824	.9631	.9412	.9206	.9053	.8986	.9023	.9166	.9398	.9690
	.9962	.9856	.9698	.9519	.9351	.9226	.9172	.9202	.9318	.9508	.9746
75°	.9974	.9899	.9788	.9661	.9542	.9453	.9415	.9436	.9518	.9652	.9821
	.9986	.9947	.9889	.9823	.9762	.9717	.9697	.9708	.9750	.9820	.9907

FIGURE 22 SOLUTION PRINTOUT FOR PROGRAM IADMC, EXAMPLE 3

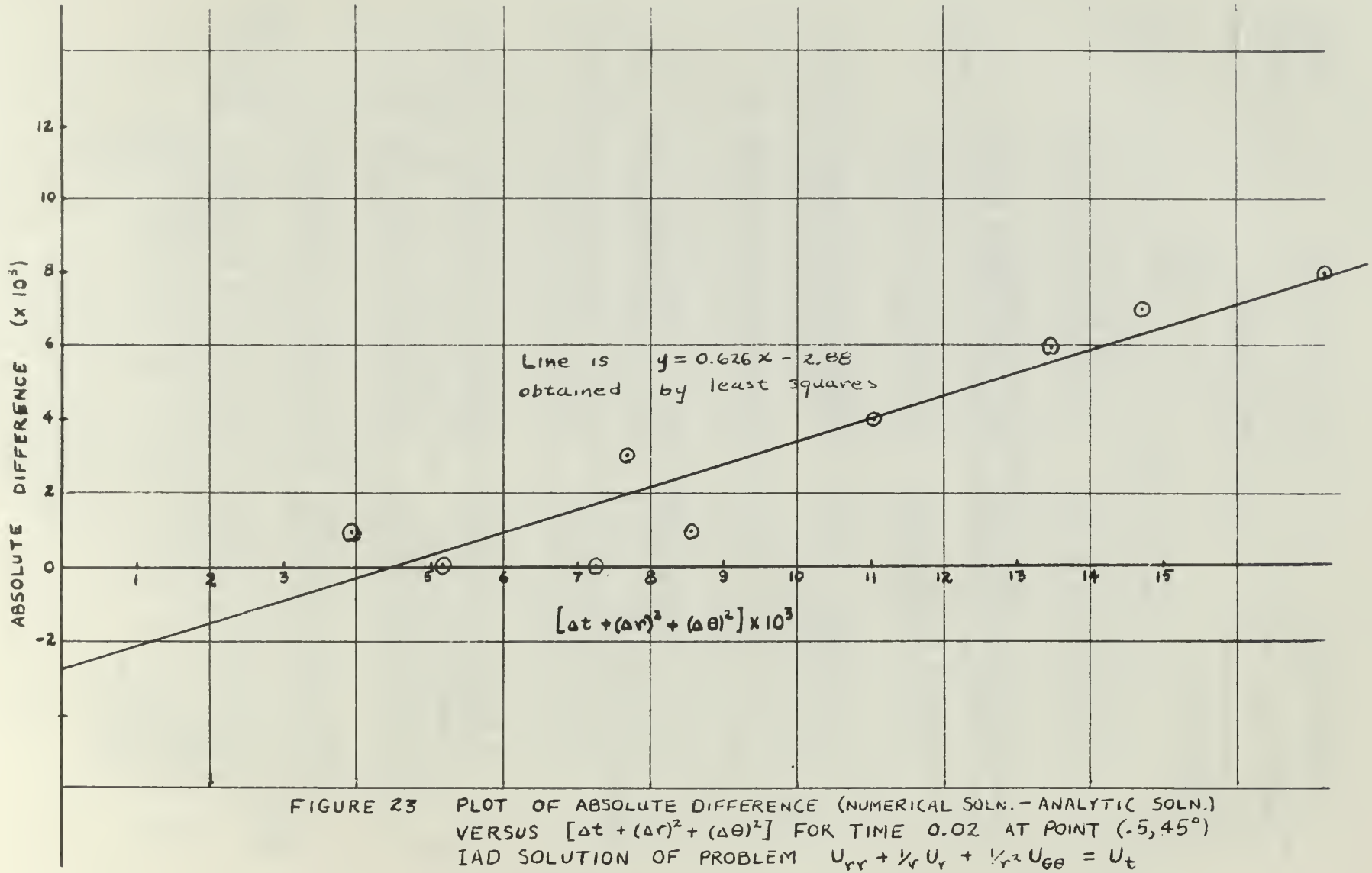

FIGURE 23 PLOT OF ABSOLUTE DIFFERENCE (NUMERICAL SOLN. − ANALYTIC SOLN.) VERSUS $[\Delta t + (\Delta r)^2 + (\Delta\theta)^2]$ FOR TIME 0.02 AT POINT (.5, 45°) IAD SOLUTION OF PROBLEM $U_{rr} + \frac{1}{r} U_r + \frac{1}{r^2} U_{\theta\theta} = U_t$

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
SOLUTION OF TWO-DIMENSIONAL HEAT PROBLEMS USING THE ALTERNATING DIRECTION METHOD

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Master of Science thesis with Major in Mathematics - September 1967

5 AUTHOR(S) *(Last name, first name, initial)*
LEIPOLD, FREDERICK J.
Lieutenant, United States Navy

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
27 September 1967	89	17

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. AVAILABILITY/LIMITATION NOTICES
[redacted]

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

Finite difference approximations to the one-dimensional heat equation $U_t = U_{xx}$ are used to introduce explicit and implicit difference equations. The convergence and stability of these equations is discussed and these concepts are used to show the restrictions imposed on explicit equations. The Implicit Alternating Direction (IAD) method is introduced as a means for solution of the two-dimensional heat equation. Although the IAD method in its basic form is applicable only to parabolic problems it is possible by slight modification to apply the method to elliptic problems. Two examples are used to illustrate the use of the IAD method for solution of parabolic and elliptic equations for a rectangular region. These examples include a work requirement comparison with other difference methods. A third example is given to show the applicability of the IAD method to non-rectangular regions, in this case a parabolic problem over a circular region. Results of these examples show that the IAD method is a convenient and accurate method when applied to both parabolic and elliptic partial differential equations and suggest applicability to a wide range of problems.

DD FORM 1 JAN 64 1473

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Finite Difference Methods						
Numerical Solution of Partial Differential Equations						

DUDLEY KNOX LIBRARY - RESEARCH REPORTS
5 6853 01077100 9
thesL477
DUDLEY KNOX LIBRARY
3 2768 00416524 1
DUDLEY KNOX LIBRARY